



ПАК ViPNet Coordinator HW

Руководство администратора

1991 – 2012 ОАО «ИнфоТеКС», Москва, Россия

ФРКЕ.00065-08 32 01, Версия 2.6

Этот документ входит в комплект поставки программного обеспечения, и на него распространяются все условия лицензионного соглашения.

Ни одна из частей этого документа не может быть воспроизведена, опубликована, сохранена в электронной базе данных или передана в любой форме или любыми средствами, такими как электронные, механические, записывающие или иначе, для любой цели без предварительного письменного разрешения ОАО «ИнфоТеКС».

ViPNet является зарегистрированной торговой маркой программного обеспечения, разрабатываемого ОАО «ИнфоТеКС».

Все торговые марки и названия программ являются собственностью их владельцев.

ОАО «ИнфоТеКС»

127287, г. Москва, Старый Петровско-Разумовский пр., дом 1/23, строение 1

Тел: (495) 737-61-96 (hotline), 737-61-92, факс 737-72-78

E-mail: hotline@infotecs.ru

WWW: <http://www.infotecs.ru>

Содержание

Введение.....	8
О документе	9
Для кого предназначен документ	9
Соглашения документа.....	9
Новые возможности	10
Что нового в версии 2.6	10
Что нового в версии 2.5	10
Что нового в версии 2.4	10
Что нового в версии 2.3	11
Что нового в версии 2.2	11
Что нового в версии 2.1	12
Что нового в версии 2.0	13
Обратная связь	15
Глава 1. Общие сведения	16
Назначение и область применения ПАК ViPNet Coordinator HW	17
Основные понятия и определения	19
Основные режимы безопасности ПО ViPNet	21
Режимы работы ПО ViPNet через межсетевой экран	23
Состав программного обеспечения.....	25
Глава 2. Аппаратная архитектура	27
Аппаратная архитектура ПАК ViPNet Coordinator HW100	28
Аппаратная архитектура ПАК ViPNet Coordinator HW1000.....	32
Аппаратная архитектура ПАК ViPNet Coordinator HW-VPNМ.....	35
Выбор консоли при загрузке ОС.....	37
Глава 3. Лицензирование ViPNet Coordinator HW	38
Лицензионные ограничения ПАК ViPNet Coordinator HW100	39
Лицензирование ПАК ViPNet Coordinator HW1000.....	40
Лицензирование ПАК ViPNet Coordinator HW-VPNМ.....	41

Глава 4. Первоначальное развертывание ключей	42
О первоначальном развертывании ключевых баз ViPNet	43
Подготовка к развертыванию ключевых баз	44
Первоначальное развертывание ключевых баз с помощью ноутбука	44
Подготовка к развертыванию ключевых баз с помощью USB-флэш	45
Процедура развертывания ключевых баз.....	46
Глава 5. Настройка ПАК с помощью командного интерпретатора.....	60
О командном интерпретаторе	61
Интерфейс командного интерпретатора	62
Средства для облегчения ввода команд	65
Сокращенный ввод команд	65
Автозаполнение	65
Контекстная подсказка	66
Команды интерпретатора.....	67
Команды группы inet	67
Команды подгруппы inet dhcp	73
Команды подгруппы inet dns.....	74
Команды подгруппы inet ntp	75
Команды группы iplir.....	76
Команды группы failover	78
Команды группы mftp	79
Команды группы admin	79
Команды группы machine	82
Команды группы ups	84
Прочие команды	85
Глава 6. Настройка конфигурации управляющего демона.....	87
Общие принципы настройки	88
Настройка параметров защищенной сети	90
Секция [id].....	90
Секция [adapter]	100
Секция [dynamic]	102
Секция [misc]	103
Секция [servers]	105
Секция [channels].....	105
Секция [service]	106

Секция [virtualip]	108
Секция [debug].....	109
Общие принципы назначения виртуальных адресов	109
Ручное переназначение виртуальных адресов узлов	111
Настройка правил обработки открытых IP-пакетов.....	112
Настройка правил антиспуфинга	114
Настройка правил фильтрации открытых IP-пакетов.....	115
Управляющий компонент.....	117
Условие	118
Действие.....	122
Расписание	122
Правила фильтрации открытых IP-пакетов по умолчанию	123
Настройка правил трансляции адресов	124
Синтаксис правил трансляции адресов.....	125
Взаимодействие правил фильтрации и правил трансляции.....	126
Служебные параметры межсетевого экрана.....	127
Настройка правил фильтрации и трансляции с помощью апплета SGA	129
Настройка параметров сетевых интерфейсов	131
Секция [mode].....	131
Секция [db].....	132
Работа с политиками безопасности	134
Настройка режимов работы через межсетевой экран	136
Настройка режима «Без использования межсетевого экрана»	136
Настройка режима «Координатор»	136
Настройка режима «Со статической трансляцией адресов».....	137
Настройка режима «С динамической трансляцией адресов»	139
Настройка работы с удаленным Координатором через фиксированный альтернативный канал.....	142
Настройка ПАК, выполняющего функции Сервера Открытого Интернета	145
Глава 7. Настройка конфигурации транспортного модуля.....	148
Назначение и функциональность транспортного модуля.....	149
Настройка параметров транспортного модуля	151
Секция [channel]	151
Специфические параметры для канала MFTR.....	152
Специфические параметры для канала SMTP.....	153
Секция [transport]	154

Секция [upgrade].....	154
Секция [mailtrans].....	155
Секция [journal]	157
Секция [misc].....	158
Секция [debug].....	160
Глава 8. Настройка системного времени	161
О настройке системного времени	162
Списки континентов, стран и временных зон.....	163
Глава 9. Удаленный мониторинг и управление ПАК.....	165
Мониторинг и управление ПАК с помощью апплета SGA	166
Настройка доступа к удаленному мониторингу и управлению	167
Глава 10. Система защиты от сбоев	169
Назначение системы защиты от сбоев.....	170
Состав системы защиты от сбоев и принципы ее работы	171
Управление системой защиты от сбоев.....	173
Настройка системы защиты от сбоев.....	174
Глава 11. Работа с конфигурациями ViPNet.....	175
О конфигурациях ViPNet.....	176
Команды для работы с конфигурациями ViPNet.....	177
Глава 12. Экспорт и импорт ключевых баз, справочников и настроек.....	180
Экспорт и импорт ключевых баз, справочников и настроек.....	181
Выполнение экспорта ключевых баз, справочников и настроек	183
Глава 13. Контроль целостности конфигурационных файлов	186
Глава 14. Сервисные функции	188
Использование ПАК в качестве DHCP-сервера	189
Использование ПАК в качестве DNS-сервера.....	191
Использование ПАК в качестве NTP-сервера	193
Взаимодействие ПАК с UPS.....	196
Глава 15. Протоколирование событий, ведение и просмотр журналов	200
Журнал регистрации IP-пакетов	201

Журнал транспортных конвертов MFTP	208
Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP	210
Журналы устранения неполадок ПО ViPNet	213
Экспорт журналов устранения неполадок ПО ViPNet.....	216
Экспорт на компьютер.....	216
Экспорт на USB-флэш	217
Глава 16. Обновление ПО ПАК ViPNet Coordinator HW	218
Удаленное обновление ПО	219
Локальное обновление ПО	220
Глава 17. Настройка работы ПАК ViPNet Coordinator HW-VPNМ.....	222
Режимы работы маршрутизатора Huawei Secoway USG.....	223
Настройка при работе маршрутизатора в режиме transparent.....	225
Настройка при работе маршрутизатора в режиме router	231
Приложение А. Примеры настройки туннелей с использованием ПАК.....	236
Приложение В. Примеры настроек работы ПАК через фиксированные альтернативные каналы	241
Приложение С. Пример использования дополнительных IP-адресов на интерфейсе	244



Введение

О документе	9
Новые возможности	10
Обратная связь	15

О документе




Для кого предназначен документ

Данный документ предназначен для администраторов, отвечающих за настройку и эксплуатацию ПО ViPNet Coordinator HW. В нем содержится как информация общего характера (назначение и применение ПАК ViPNet Coordinator HW, аппаратные конфигурации и лицензирование), так и описание конкретных действий администратора по настройке и управлению ПАК.

Соглашения документа

Соглашения данного документа представлены в таблице ниже.

Таблица 1. Условные обозначения

Указатель	Описание
	Внимание! Указывает на обязательное для исполнения или следования действие или информацию.
	Примечание. Указывает на необязательное, но желательное для исполнения или следования действие или информацию.
	Совет. Содержит дополнительную информацию общего характера.

НОВЫЕ ВОЗМОЖНОСТИ

Что нового в версии 2.6

В этом разделе представлен краткий обзор изменений версии 2.6.

- **Исправление ошибок в программном обеспечении**

Исправлены следующие ошибки, проявившиеся при эксплуатации ПАК ViPNet Coordinator HW100 базовой конфигурации:

- Невозможно установить ПО, установка завершается ошибкой.
- Не запускается транспортный модуль MFTR.

Что нового в версии 2.5

В этом разделе представлен краткий обзор изменений и новых возможностей версии 2.5.

- **Контроль конфигурации дисковой подсистемы при установке ПО на ПАК ViPNet Coordinator HW1000 G2**

Доработана программа установки ПО на загрузочный носитель ПАК. Необходимость доработки обусловлена появлением дополнительной конфигурации ПАК ViPNet Coordinator HW1000 G2 с нестандартным подключением дисков. Теперь при установке ПО на эту модификацию анализируется конфигурация дисковой подсистемы. По результатам анализа выводится информация о подключенных дисковых накопителях и запрашивается подтверждение на продолжение установки. В случае обнаружения некорректной конфигурации установка автоматически прекращается с выводом соответствующего сообщения.

Что нового в версии 2.4

В этом разделе представлен краткий обзор изменений и новых возможностей версии 2.4.

- **Возможность ограничения диапазона генерируемых виртуальных адресов**

Реализована возможность ограничить диапазон генерируемых виртуальных адресов. Для этого используется новый параметр `maxvirtualip`, задаваемый в секции `[virtualip]` файла конфигурации `iplir.conf` (см. «Секция [\[virtualip\]](#)» на стр. 108).

Теперь генерируемые виртуальные адреса не могут превышать значения, заданного этим параметром.

Что нового в версии 2.3

В этом разделе представлен краткий обзор изменений и новых возможностей версии 2.3.

- **Поддержка кластера горячего резервирования на базе ПАК ViPNet Coordinator HW-VPNM**

Реализована поддержка работы кластера горячего резервирования, организованного на базе ПАК ViPNet Coordinator HW-VPNM.

- **Поддержка взаимодействия ПАК ViPNet Coordinator HW с UPS**

Реализована поддержка взаимодействия ПАК ViPNet Coordinator HW с источниками бесперебойного питания (UPS). Теперь ПАК, получив от UPS сигнал об истощении батареи, корректно завершает свою работу. Настройка и управление взаимодействием ПАК с UPS производится с помощью командного интерпретатора.

Что нового в версии 2.2

В этом разделе представлен краткий обзор изменений и новых возможностей версии 2.2.

- **Расширенная поддержка системы централизованного мониторинга ViPNet StateWatcher**

Реализованы команды для передачи расширенной информации о состоянии ПАК в систему централизованного мониторинга. Теперь в систему мониторинга дополнительно передается информация о работоспособности транспортного модуля MFTR, количество конвертов в очереди и их суммарный размер, список туннелируемых ПАК адресов, суммарный трафик на каждом сетевом интерфейсе (отдельно исходящий и входящий), загрузка процессора, использование памяти и дискового пространства.

- **Изменен поддерживаемый SSH-протокол**

SSH-сервер, встроенный в ПАК, переключен на поддержку протокола SSH2. Протокол SSH1 больше не поддерживается.

- **Поддержка протокола SCCP**

Реализована поддержка расширения Skinny при обработке протокола SCCP.

Что нового в версии 2.1

В этом разделе представлен краткий обзор изменений и новых возможностей версии 2.1.

- **Пополнился список поддерживаемых поколений ПАК ViPNet Coordinator HW100**

Для ПАК ViPNet Coordinator HW100 реализована поддержка новой аппаратной платформы на базе компактного компьютера BK3741S-00C серии BRIK. Теперь предыдущая модификация ПАК на базе компьютера серии eBox-4 относится к первому поколению (G1), а модификация на новой платформе — ко второму поколению (G2).

- **Пополнился список поддерживаемых поколений ПАК ViPNet Coordinator HW1000**

Для ПАК ViPNet Coordinator HW1000 реализована поддержка новой аппаратной платформы на базе сервера AquaServer T40 S44. Теперь предыдущая модификация ПАК на базе сервера T40 S42 относится к первому поколению (G1), а модификация на новой платформе — ко второму поколению (G2).

- **Изменены правила фильтрации по умолчанию**

Правила фильтрации открытых IP-пакетов, заданные по умолчанию, а также правила режимов безопасности приведены в соответствие с версией ПО ViPNet Coordinator для ОС Windows. Теперь ПАК имеют такую же логику поведения, что и координаторы, работающие под управлением ОС Windows.

- **Изменены правила антиспуфинга**

Правила антиспуфинга приведены в соответствие с версией ПО ViPNet Coordinator для ОС Windows. Теперь правила антиспуфинга не зависят от типа интерфейса, а в качестве допустимых адресов отправителя можно указывать комбинацию адресов.

- **Поддержка правил фильтрации туннелируемого трафика**

Реализована поддержка правил фильтрации туннелируемого трафика. Теперь эти правила задаются в отдельной секции [tunnel] файла конфигурации `iplir.conf`. Как следствие, упразднен параметр `autopasstunnels` в секции [misc] файла конфигурации `iplir.conf`.

- **Возможность указания типа и кода ICMP-пакетов в правилах фильтрации**

Реализована поддержка типа и кода ICMP-пакетов в правилах фильтрации открытых IP-пакетов. Теперь тип и код ICMP-пакетов можно указать как непосредственно в файле конфигурации, так и с помощью апплета мониторинга и управления SGA.

- **Поддержка различных сервисных функций**

В состав ПАК ViPNet Coordinator HW включены DHCP-, NTP- и DNS-серверы с возможностью настройки и управления с помощью командного интерпретатора.

- **Изменена реализация процедуры развертывания ключевых баз ViPNet**
Процедура развертывания ключевых баз ViPNet реализована в виде мастера, который позволяет в консольном или псевдо-графическом режиме произвести не только развертывание ключевых баз, но и дополнительные настройки.
- **Возможность установки временной зоны и времени**
Реализована возможность установки временной зоны (часового пояса) и текущего времени. Установку можно произвести как в процессе развертывания ключевых баз ViPNet, так и с помощью командного интерпретатора.
- **Поддержка многопоточности в драйвере ViPNet**
Реализована поддержка многопоточности в драйвере ViPNet. Теперь можно управлять числом потоков с помощью специальной команды. Для совместимости с предыдущими версиями по умолчанию установлен однопоточный режим.
- **Поддержка дополнительных IP-адресов на сетевых интерфейсах**
Реализована возможность задания дополнительных IP-адресов на сетевых интерфейсах ПАК с помощью командного интерпретатора.

Что нового в версии 2.0

В этом разделе представлен краткий обзор изменений и новых возможностей версии 2.0.

- **Контроль целостности конфигурационных файлов**
Реализован контроль целостности конфигурационных файлов ОС Linux и служб, входящих в состав ПАК ViPNet Coordinator HW. Проверка целостности конфигурационных файлов выполняется при каждой попытке их использования.
- **Расширенные возможности настройки с помощью апплета SGA**
Реализована поддержка настройки фильтров открытой сети, правил трансляции адресов и режимов безопасности сетевых интерфейсов с помощью апплета SGA, а также поддержка авторизации доступа с помощью SGA.
- **Ограничение числа попыток ввода пароля**
Реализован контроль числа попыток ввода пароля: теперь допускается не более 10-и попыток. После 10-и неудачных попыток ввода пароля ПАК перезагружается.
- **Регламентное тестирование ПАК**

Реализована процедура регламентного тестирования, которая автоматически выполняется при старте ПАК, а также может быть запущена вручную с помощью соответствующей команды.

- **Возможность локального ведения и экспорта журналов устранения неполадок ПО ViPNet**

Реализовано локальное ведение журналов устранения неполадок ПО ViPNet на модификациях ПАК с жестким диском, а также экспорт журналов на внешний компьютер или USB-флэш.

- **Возможность экспорта ключевых баз, справочников и настроек на USB-флэш**

Реализован экспорт ключевых баз, справочников и настроек на USB-флэш в дополнение к экспорту на внешний компьютер.

- **Возможность локального обновления ПО**

Реализовано локальное обновление ПО с USB-флэш, которое выполняется с помощью соответствующей команды.

- **Поддержка работы ПАК ViPNet Coordinator HW100 с SATA HDD SeaGate**

Реализована поддержка функционирования ПАК ViPNet Coordinator HW100 расширенной конфигурации с SATA HDD SeaGate.

Обратная связь

Дополнительная информация

Сведения о продуктах и решениях ViPNet, распространенные вопросы и другая полезная информация собраны на сайте компании «ИнфоТеКС». По предложенным ссылкам можно найти ответы на многие вопросы, возникающие в процессе эксплуатации продуктов ViPNet.

- Сборник часто задаваемых вопросов (FAQ) <http://www.infotecs.ru/support/faq/>.
- Законодательная база в сфере защиты информации <http://www.infotecs.ru/laws/>.
- Информация о решениях ViPNet <http://www.infotecs.ru/solutions/vpn/>.

Контактная информация

С вопросами по использованию продуктов ViPNet, пожеланиями или предложениями свяжитесь со специалистами компании «ИнфоТеКС». Для решения возникающих проблем обратитесь в службу технической поддержки.

- Электронный адрес службы поддержки hotline@infotecs.ru.
- Форма запроса по электронной почте в службу поддержки <http://www.infotecs.ru/support/request/>.
- Форум компании «ИнфоТеКС» <http://www.infotecs.ru/forum>.
- 8 (495) 737-6196 — «горячая линия» службы поддержки.
- 8 (800) 250-0260 — бесплатный звонок из любого региона России (кроме Москвы).



1

Общие сведения

Назначение и область применения ПАК ViPNet Coordinator HW	17
Основные понятия и определения	19
Основные режимы безопасности ПО ViPNet	21
Режимы работы ПО ViPNet через межсетевой экран	23
Состав программного обеспечения	25

Назначение и область применения ПАК ViPNet Coordinator HW

Программно-аппаратный комплекс ViPNet Coordinator HW (далее ПАК ViPNet Coordinator HW или ПАК) представляет собой интегрированное решение на базе нескольких аппаратных платформ и программного обеспечения производства ОАО «ИнфоТеКС», предназначенное для организации сетевой защиты в VPN-сетях. В качестве аппаратной платформы в ПАК может использоваться компактный компьютер или полноценный сервер, устанавливаемый в стандартные стойки.

В зависимости от используемой аппаратной платформы различают следующие модификации ПАК ViPNet Coordinator HW:

- ПАК ViPNet Coordinator HW100 – ПАК на базе компактных безвентиляторных компьютеров, поставляемый на двух аппаратных платформах (см. [«Аппаратная архитектура ПАК ViPNet Coordinator HW100»](#) на стр. 28).
- ПАК ViPNet Coordinator HW1000 – ПАК на базе телеком-серверов серии AquaServer T, поставляемый на двух аппаратных платформах (см. [«Аппаратная архитектура ПАК ViPNet Coordinator HW1000»](#) на стр. 32).
- ПАК ViPNet Coordinator HW-VPNМ – ПАК на базе модуля расширения VPNМ для маршрутизаторов серии Secoway USG (см. [«Аппаратная архитектура ПАК ViPNet Coordinator HW-VPNМ»](#) на стр. 35).

В состав всех модификаций ПАК входит программное обеспечение (ПО) ViPNet Coordinator Linux, которое является одним из программных продуктов семейства ViPNet CUSTOM Linux и обеспечивает следующую основную функциональность ПАК:

- Криптошлюза для организации защищенных туннелей в рамках виртуальной частной сети ViPNet.
- Межсетевого экрана.
- Сервера IP-адресов виртуальной частной сети ViPNet (поддержка работы удаленных мобильных пользователей и любых других узлов сети с динамическими IP-адресами).
- Сервера-маршрутизатора почтовых конвертов.

- Сервера Открытого Интернета для организации безопасного подключения к Интернет отдельных узлов сети ViPNet без их физического отключения от локальной сети.

ПАК ViPNet Coordinator HW100, как решение на базе мини-компьютеров, имеет ограничение по функциональности: функция Сервера-маршрутизатора обеспечивается только при комплектации компьютера дополнительным жестким диском (см. [«Лицензионные ограничения ПАК ViPNet Coordinator HW100»](#) на стр. 39).

Для создания отказоустойчивого решения в ПО ViPNet Coordinator Linux реализована технология горячего резервирования ПАК, объединенных в кластер. Эта технология применима только к ПАК ViPNet Coordinator HW1000 и HW-VPNM. В случае выхода из строя одного из ПАК, входящих в кластер, переключение на второй (резервный) ПАК происходит автоматически, без вмешательства администратора. Подробное описание кластера горячего резервирования содержится в документе «ПАК ViPNet Coordinator HW. Система защиты от сбоев. Руководство администратора».

ПО ViPNet Coordinator Linux в составе ПАК функционирует под управлением адаптированной ОС Linux.

Основные понятия и определения

В данном разделе приведены основные понятия и определения, используемые в технологии построения виртуальных частных сетей ViPNet.

Виртуальная частная сеть ViPNet – сеть, состоящая из компьютеров, на которых установлено ПО ViPNet. Каждая сеть ViPNet имеет свой уникальный номер при поставке сети конкретному заказчику. Номер сети присутствует в идентификаторах объектов сети в виде четырехзначного шестнадцатеричного числа.

Каждая сеть ViPNet должна содержать свой **Центр управления сетью (ЦУС)** и свой **Удостоверяющий и ключевой центр (УКЦ)**. ЦУС и УКЦ выполняют административные функции и отвечают за конфигурирование сети, возможность организации защищенных связей между объектами сети, генерацию ключей, используемых для шифрования, и так далее. Сеть ViPNet также может содержать **Центр управления политиками безопасности (ЦУПБ)**, который отвечает за формирование корпоративной политики безопасности и ее рассылку на сетевые узлы.

Сеть ViPNet состоит из **сетевых узлов (СУ)**, каждый из которых является либо **Координатором** (с установленным ПО ViPNet Coordinator), либо **Клиентом** (с установленным ПО ViPNet Client).

ПАК ViPNet Coordinator HW является одним из элементов (сетевых узлов) виртуальной сети ViPNet и выполняет функции Координатора сети. Координаторы отличаются от Клиентов тем, что могут выполнять ряд дополнительных функций:

- **Функция Сервера IP-адресов.**

Координаторы хранят информацию об адресах Клиентов. Когда включается какой-либо Клиент, он посылает информацию о своем включении одному из Координаторов (который является для этого Клиента его Сервером IP-адресов), а Координатор сообщает эту информацию другим Клиентам, так что все Клиенты имеют сведения об адресах друг друга. Если в сети несколько Координаторов, то они обмениваются этой информацией между собой.

- **Функция межсетевого экрана.**

Координаторы осуществляют фильтрацию открытых пакетов в соответствии с заданной политикой безопасности.

- **Функция прокси-сервера.**

Координаторы могут выступать в качестве прокси-серверов для Клиентов. При этом другим Клиентам не известен реальный адрес Клиента, а известен только адрес его прокси-сервера. Для связи с таким Клиентом используется виртуальный адрес.

- **Функция Сервера-маршрутизатора.**

Координаторы выполняют маршрутизацию почтовых конвертов и управляющих сообщений при взаимодействии узлов сети между собой.

- **Функция туннелирования трафика.**

Координаторы могут туннелировать трафик от открытых компьютеров сети (например таких, на которые по каким-либо причинам невозможно установить ПО ViPNet). При этом трафик остается незащищенным только на участке от туннелируемого компьютера до Координатора, а в дальнейшем весь трафик идет в зашифрованном виде.

- **Функция Сервера Открытого Интернета.**

Координаторы могут использоваться для организации безопасного подключения к Интернету отдельных узлов сети ViPNet без их физического отключения от локальной сети.

Основные режимы безопасности ПО ViPNet

ПО ViPNet может работать в нескольких режимах безопасности. Каждый из этих режимов представляет собой определенный набор правил, по которым производится обработка зашифрованных и незашифрованных пакетов. Режим работы ПО ViPNet в ПАК ViPNet Coordinator HW устанавливается отдельно для каждого сетевого интерфейса, что позволяет производить гибкую настройку в соответствии со способом использования ПАК.

Существуют следующие режимы безопасности:

1 Блокировать IP-пакеты всех соединений

Этот режим – самый жесткий и безопасный, он эквивалентен физическому отключению компьютера от открытых источников внешней сети. В данном режиме возможно только защищенное взаимодействие с сетевыми узлами. Установка этого режима на каком-либо сетевом интерфейсе ПАК приведет к полной блокировке как входящих, так и исходящих открытых IP-пакетов на этом интерфейсе независимо от разрешающих сетевых фильтров.

2 Блокировать все соединения, кроме разрешенных

В этом режиме обеспечивается защищенное взаимодействие с сетевыми узлами. Для открытых IP-пакетов установка данного режима на каком-либо сетевом интерфейсе ПАК обеспечивает пропуск открытых IP-пакетов, явно разрешенных сетевыми фильтрами. Остальной открытый трафик блокируется.

3 Пропускать все исходящие соединения, кроме запрещенных

Этот режим предназначен для обеспечения более безопасной работы в Интернете с открытыми серверами и другими компьютерами. В данном режиме обеспечивается защищенное взаимодействие с сетевыми узлами. Для открытых IP-пакетов установка этого режима для какого-либо сетевого интерфейса ПАК обеспечивает блокировку пакетов, явно запрещенных сетевыми фильтрами, и пропуск пакетов, явно разрешенных сетевыми фильтрами. Для всех остальных открытых IP-пакетов, поступающих на сетевой интерфейс, работает следующее правило (правило бумеранга):

- Через сетевой интерфейс разрешаются инициативные исходящие соединения, при этом драйвер ViPNet запоминает, на какой сетевой ресурс (по какому IP-

адресу, протоколу и номеру порта) происходит обращение, и обратно IP-пакеты пропускаются только с этого ресурса.

- Весь остальной IP-трафик, в том числе с того же IP-адреса, но с другими параметрами (тип протокола, номер порта), будет заблокирован.
- Если в течение определенного промежутка времени (который зависит от типа протокола) после инициации соединения с данного сетевого ресурса не поступил ответный IP-пакет, ПО ViPNet автоматически убирает разрешение на прохождение трафика, что приводит к невозможности атаки с указанного ресурса впоследствии.

4 Пропускать все соединения

В этом режиме обеспечивается защищенное взаимодействие с сетевыми узлами. При установке данного режима на каком-либо сетевом интерфейсе ПАК будут пропускаться все открытые IP-пакеты (как входящие, так и исходящие) независимо от запрещающих сетевых фильтров. На интерфейсе, обеспечивающем доступ во внешнюю сеть (например, в Интернет), этот режим рекомендуется использовать только кратковременно при отладочных работах.

5 Пропускать IP-пакеты без обработки

Этот режим полностью отключает работу драйвера ViPNet на данном сетевом интерфейсе ПАК. При этом защищенные пакеты не расшифровываются, поэтому защищенное взаимодействие с сетевыми узлами невозможно. Использовать этот режим не рекомендуется, за исключением специальных случаев, например, при организации управляющего канала в системе горячего резервирования. Как правило, такие случаи оговариваются в документации отдельно.

Правила режимов безопасности 2, 3 и 4 применяются только к локальному и широкополосному открытому трафику, они не распространяются на транзитный и туннелируемый трафик. Транзитные и туннелируемые IP-пакеты в указанных режимах блокируются, если они явно не разрешены сетевыми фильтрами.



Примечание. По умолчанию на всех интерфейсах ПАК устанавливается режим безопасности 2.

ПО ViPNet является в большой степени самонастраивающейся системой. В процессе своей работы управляющая программа ViPNet посылает специальные защищенные пакеты, которые позволяют компьютерам с этим ПО увидеть друг друга в локальной сети, сообщить друг другу о своих адресах, если они изменились. Чтобы компьютеры увидели друг друга в разветвленной сети, в ней должны функционировать несколько компьютеров, выполняющих функции Сервера IP-адресов.

Режимы работы ПО ViPNet через межсетевой экран

Для обеспечения возможности работы ПАК при разных способах его подключения к внешней сети в ПО ViPNet предусмотрено четыре режима работы через межсетевой экран:

- 1 Без использования межсетевого экрана** – автономная работа ПАК без использования внешнего межсетевого экрана.
- 2 Координатор** – работа ПАК через другой Координатор (при каскадной схеме установки Координаторов).
- 3 Со статической трансляцией адресов** – работа ПАК через внешний межсетевой экран (устройство) с трансляцией адресов (NAT), на котором возможна настройка статических правил трансляции адресов.
- 4 С динамической трансляцией адресов** – работа ПАК через внешний межсетевой экран (устройство), на котором осуществляется динамическая трансляция адресов (наиболее распространенный способ подключения к WAN).

Если ПАК имеет непосредственное подключение к внешней сети, т.е. может быть доступен напрямую со стороны любых других сетевых узлов (например, имеет публичный адрес), то для него следует выбрать режим **Без использования межсетевого экрана**.

Если ПАК имеет частный IP-адрес, по которому нельзя получить доступ со стороны некоторых других сетевых узлов в соответствии с общими правилами маршрутизации (то есть на выходе во внешнюю сеть установлен межсетевой экран или иное устройство, выполняющее преобразование адресов), то необходимо выбрать один из режимов работы через межсетевой экран.

Режим **Координатор** выбирается в случае, если на выходе во внешнюю сеть уже установлен другой Координатор, выполняющий функции межсетевого экрана. В этом случае установленный Координатор выбирается в качестве межсетевого экрана для ПАК.

Режим **Со статической трансляцией адресов** выбирается в случае, если на выходе во внешнюю сеть установлен межсетевой экран или иное NAT-устройство, на котором можно настроить статические правила трансляции адресов, обеспечивающие

взаимодействие с определенным внутренним адресом сети по протоколу UDP с заданным портом.

Режим **С динамической трансляцией адресов** выбирается в случае, если на выходе во внешнюю сеть установлен межсетевой экран или иное NAT-устройство, на котором затруднительно настроить статические правила трансляции адресов. Следует отметить, что данный режим наиболее универсален и ПАК в этом режиме будет работоспособен и при других способах подключения.



Примечание. Если узлы находятся в одной локальной сети в области доступности друг друга по широковещательным пакетам, то независимо от выбранного режима работы взаимодействие между ними всегда осуществляется напрямую по IP-адресу узла.

Подробное описание настройки каждого из режимов приведено в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136).

В любом из описанных режимов работы ПАК через межсетевой экран можно установить каждый его сетевой интерфейс в любой из режимов безопасности, описанных выше (см. «[Основные режимы безопасности ПО VipNet](#)» на стр. 21).

Состав программного обеспечения

В состав ПО ПАК ViPNet Coordinator HW входят следующие основные функциональные модули:

- **Низкоуровневый драйвер сетевой защиты `iplir`**, взаимодействующий непосредственно с драйверами сетевых карт и контролирующий весь обмен трафиком данного компьютера с внешней сетью.
- **Управляющая программа-демон `iplircfg`**, которая осуществляет загрузку необходимых параметров драйверу `iplir`, рассылку и прием информации об IP-адресах клиентов, ведение журнала трафика и т.п. (см. «[Настройка конфигурации управляющего демона](#)» на стр. 87). Рекомендуется, чтобы эта программа всегда была запущена, но при ее остановке драйвер `iplir` продолжает работать и обмен трафиком не прерывается.
- **Криптографический драйвер**, выполняющий криптографические операции по запросу драйвера `iplir`.
- **Драйвер `watchdog`**, входящий в состав системы защиты от сбоев (см. «[Состав системы защиты от сбоев и принципы ее работы](#)» на стр. 171). Драйвер работает на очень низком уровне и в большинстве случаев сохраняет работоспособность даже тогда, когда система уже не реагирует на внешние события.
- **Демон `failoverd`**, который обеспечивает функционал системы защиты от сбоев (см. «[Система защиты от сбоев](#)» на стр. 169).
- **Демон `mftpd` (транспортный модуль MFTR)**, который обеспечивает прием и передачу транспортных конвертов между узлами сети ViPNet (см. «[Настройка конфигурации транспортного модуля](#)» на стр. 148).
- **SNMP-демон**, который позволяет получать статистику работы ПО ViPNet с удаленных хостов по протоколу SNMP (см. «[Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP](#)» на стр. 210).
- **Командный интерпретатор**, с помощью которого осуществляется администрирование ПАК (см. «[Настройка ПАК с помощью командного интерпретатора](#)» на стр. 60).
- **Веб-сервер Apache с апплетом SGA (Security Gateway Applet)**, обеспечивающий мониторинг и управление ПАК посредством веб-интерфейса. Описание апплета

SGA содержится в документе «Апплет мониторинга и управления ViPNet-координатором. Руководство пользователя», входящем в комплект поставки.

- **DHCP-сервер**, который позволяет динамически назначать IP-адреса сетевым узлам (см. «[Использование ПАК в качестве DHCP-сервера](#)» на стр. 189).
- **DNS-сервер**, обеспечивающий разрешение символьных имен в IP-адреса (см. «[Использование ПАК в качестве DNS-сервера](#)» на стр. 191).
- **NTP-сервер**, с помощью которого осуществляется синхронизация времени (см. «[Использование ПАК в качестве NTP-сервера](#)» на стр. 193).
- **Пакет NUT (Network UPS Tools)**, обеспечивающий взаимодействие ПАК с источником бесперебойного питания.



2

Аппаратная архитектура

Аппаратная архитектура ПАК ViPNet Coordinator HW100	28
Аппаратная архитектура ПАК ViPNet Coordinator HW1000	32
Аппаратная архитектура ПАК ViPNet Coordinator HW-VPNМ	35
Выбор консоли при загрузке ОС	37

Аппаратная архитектура ПАК ViPNet Coordinator HW100

В качестве аппаратной платформы в ПАК ViPNet Coordinator HW100 используются мини-компьютеры с пассивным охлаждением (без вентилятора охлаждения), с низким уровнем тепловыделения и энергопотребления. Компьютеры имеют компактные габаритные размеры и небольшой вес, их применение особенно оправдано в тех местах, где физическое пространство ограничено, а условия окружающей среды неблагоприятны.

ПАК ViPNet Coordinator HW100 представлен двумя поколениями, которые различаются используемой аппаратной платформой:

- в первом поколении ПАК ViPNet Coordinator HW100 (ПАК HW100 G1) в качестве аппаратной платформы используется компьютер серии eBox-4;
- во втором поколении ПАК ViPNet Coordinator HW100 (ПАК HW100 G2) в качестве аппаратной платформы используется компьютер BK3741S-00C серии BRIK, производимый компанией «Lex Computech».

На рисунке ниже представлен внешний вид ПАК HW100 G1.



Рисунок 1: Внешний вид ПАК HW100 G1

ПАК HW100 G1 имеет следующие характеристики:

Таблица 2. Характеристики ПАК HW100 G1

Характеристика	Описание
Процессор	VIA Eden Esther с частотой 1.2 ГГц, совместим с i386
Оперативная память	512 МБ
Электронный диск	Compact Flash 512 МБ (флэш-диск)
Сетевые интерфейсы	2 интерфейса Ethernet Realtek RTL8139 10/100
Графический контроллер	VGA, объем видеопамяти 64 МБ
COM	1 порт RS-232
USB	2 порта Rev. 2.0
Дополнительные интерфейсы	PS/2 Клавиатура, PS/2 Мышь
Мощность источника питания	25 Вт (внешний адаптер AC-DC 220В AC/5В DC)

Один разъем USB расположен на передней панели компьютера, остальные коммуникационные разъемы находятся на задней панели компьютера.

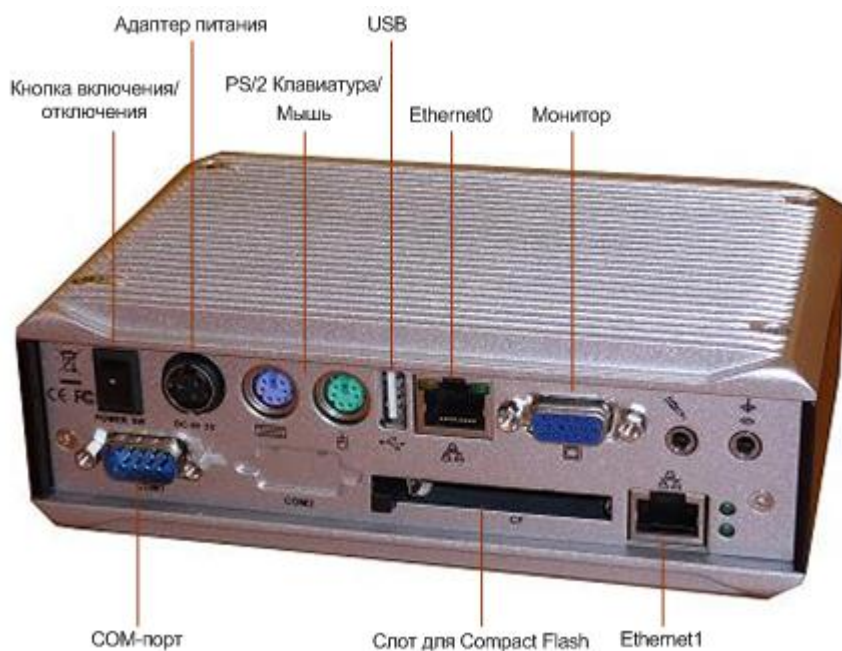


Рисунок 2: Задняя панель ПАК HW100 G1

На рисунке ниже представлен внешний вид ПАК HW100 G2.



Рисунок 3: Внешний вид ПАК HW100 G2

ПАК HW100 G2 имеет следующие характеристики:

Таблица 3. Характеристики ПАК HW100 G2

Характеристика	Описание
Процессор	Intel Atom N270 с частотой 1.6 ГГц
Оперативная память	1 ГБ
Электронный диск	Compact Flash 1 ГБ (флэш-диск)
Сетевые интерфейсы	4 интерфейса Ethernet Realtek 8111C 10/100/1000 Мбит/с
Графический контроллер	VGA
USB	2 порта Rev. 2.0
Мощность источника питания	12 Вт (внешний адаптер 12В AC-DC)

Все коммуникационные разъемы расположены на задней панели компьютера.

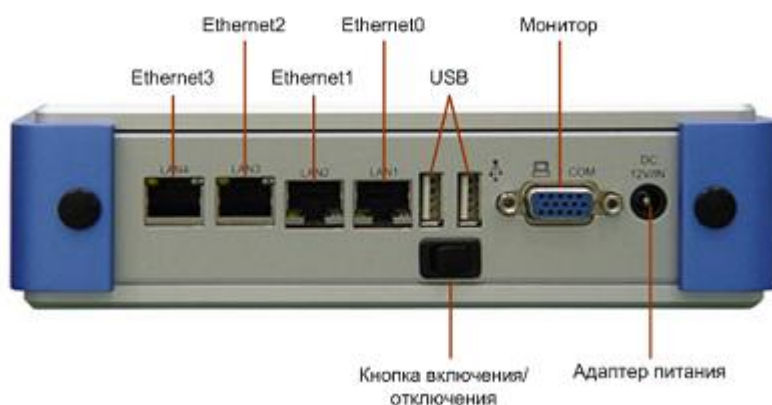


Рисунок 4: Задняя панель ПАК HW100 G2

Компьютеры, используемые в качестве аппаратной платформы, могут дополнительно комплектоваться жестким диском, что позволяет расширить функциональность ПАК (см. «[Лицензионные ограничения ПАК ViPNet Coordinator HW100](#)» на стр. 39). В связи с этим различают 2 конфигурации ПАК ViPNet Coordinator HW100:

- 1 Базовая конфигурация** – в качестве диска используется только флэш-диск (Compact Flash).
- 2 Расширенная конфигурация** – флэш-диск дополняется жестким диском (HDD).

На флэш-диске установлено ПО ViPNet Coordinator Linux, которое функционирует под управлением адаптированной ОС Linux.

В качестве консоли можно использовать следующее оборудование:

- ноутбук, подключенный к COM-порту (COM-консоль) – только для ПАК HW100 G1;
- монитор и клавиатуру (обычная консоль).

Перед началом эксплуатации ПАК на нем должны быть развернуты ключевые базы ViPNet. При развертывании ключевых баз используется мобильный компьютер (ноутбук), подключенный к порту Ethernet ПАК, или USB-флэш, вставленный в USB-разъем ПАК (см. «[Первоначальное развертывание ключей](#)» на стр. 42). Управление процедурой развертывания можно осуществлять с помощью подключения по Telnet или с одной из консолей.

После успешного развертывания ключевых баз ViPNet подключиться к ПАК по Telnet невозможно. Дальнейшее администрирование ПАК осуществляется локально с одной из возможных консолей и/или удаленно по протоколу SSH (см. «[Настройка ПАК с помощью командного интерпретатора](#)» на стр. 60).

Аппаратная архитектура ПАК ViPNet Coordinator HW1000

ПАК ViPNet Coordinator HW1000 базируется на телеком-серверах серии AquaServer T производства ГК «Аквариус». Эта модификация ПАК представлена двумя поколениями, которые различаются используемой аппаратной платформой:

- в первом поколении ПАК ViPNet Coordinator HW1000 (ПАК HW1000 G1) в качестве аппаратной платформы используется сервер AquaServer T40 S42;
- во втором поколении ПАК ViPNet Coordinator HW1000 (ПАК HW1000 G2) в качестве аппаратной платформы используется сервер AquaServer T40 S44.

В ПАК HW1000 G1 в качестве накопителей используются два жестких диска, объединенных в RAID-массив. Во втором поколении используются обычный жесткий диск (HDD) и встроенный флэш-модуль SSD. RAID-массив во втором поколении не поддерживается.



Внимание! ПАК HW1000 G1 работает на аппаратной конфигурации только с двумя жесткими дисками, объединенными в RAID-массив.

ПАК HW1000 G1 обладает конструктивной особенностью корпуса – его глубина составляет половину стандартной модели. В стандартной 19” стойке на одном уровне могут быть установлены два таких ПАК. Применение ПАК HW1000 G1 хорошо подходит для использования в решениях с большой аппаратной нагрузкой на единицу площади.

ПАК HW1000 G1 имеет следующие технические характеристики:

Таблица 4. Характеристики ПАК HW1000 G1

Характеристика	Описание
Процессор	Intel Core 2 Duo
Количество ядер	2
Оперативная память	2 ГБ
Поддержка RAID	Программная поддержка на уровне BIOS

На передней панели ПАК HW1000 G1 расположены 2 разъема USB, остальные коммуникационные разъемы находятся на задней панели.

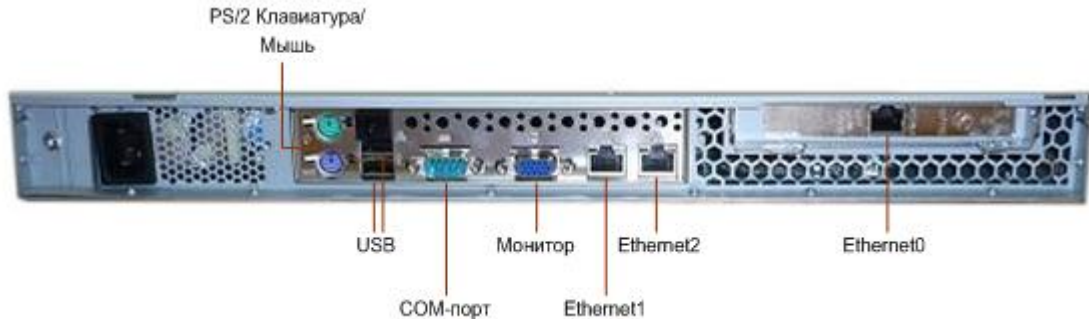


Рисунок 5: Задняя панель ПАК HW1000 G1

ПАК HW1000 G2 имеет следующие технические характеристики:

Таблица 5. Характеристики ПАК HW1000 G2

Характеристика	Описание
Процессор	Intel Core i3-530
Количество ядер	2
Оперативная память	2 x 1024 МБ
SSD	2 Гб Sata 2.5''
HDD	от 250 Гб
Поддержка RAID	Не поддерживается
Сетевые интерфейсы	4 интерфейса Ethernet 10/100/1000 Мбит/с RJ45

На передней панели ПАК HW1000 G2 расположены 2 разъема USB, остальные коммуникационные разъемы находятся на задней панели.

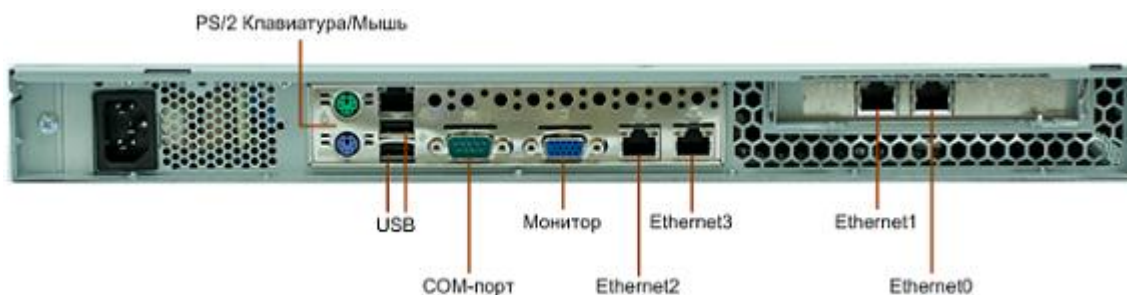


Рисунок 6: Задняя панель ПАК HW1000 G2

В качестве ОС на ПАК ViPNet Coordinator HW1000 используется адаптированная ОС Linux.

В качестве консоли можно использовать следующее оборудование:

- ноутбук, подключенный к COM-порту (COM-консоль);
- монитор и клавиатуру (обычная консоль).

Перед началом эксплуатации ПАК на нем должны быть развернуты ключевые базы ViPNet. При развертывании ключевых баз используется мобильный компьютер (ноутбук), подключенный к порту Ethernet ПАК, или USB-флэш, вставленный в USB-разъем ПАК (см. [«Первоначальное развертывание ключей»](#) на стр. 42). Управление процедурой развертывания можно осуществлять с помощью подключения по Telnet или с одной из консолей.

После успешного развертывания ключевых баз ViPNet подключиться к ПАК по Telnet невозможно. Дальнейшее администрирование комплекса осуществляется локально с одной из возможных консолей и/или удаленно по протоколу SSH (см. [«Настройка ПАК с помощью командного интерпретатора»](#) на стр. 60).

Аппаратная архитектура ПАК ViPNet Coordinator HW-VPNM

В качестве аппаратной платформы в ПАК ViPNet Coordinator HW-VPNM используется модуль расширения VPNM (VPN Module) для маршрутизаторов серии Secoway USG, производимых компанией Huawei Symantec.



Рисунок 7: Внешний вид модуля VPNM

Модуль VPNM имеет следующие характеристики:

Таблица 6. Характеристики модуля расширения VPNM

Характеристика	Описание
Процессор	Intel Core 2 Duo T7500 с частотой 2.2 ГГц
Оперативная память	1 ГБ
Электронный диск	Compact Flash 1 ГБ (флэш-диск)
HDD	160 ГБ
Сетевые интерфейсы	1 внутренний интерфейс 1000base-FX для подключения модуля к маршрутизатору 2 интерфейса 10/100/1000base-TX
Графический контроллер	VGA
COM	1 порт
USB	2 порта Rev. 2.0

Все коммуникационные разъемы (кроме внутреннего интерфейса) находятся на лицевой панели модуля.

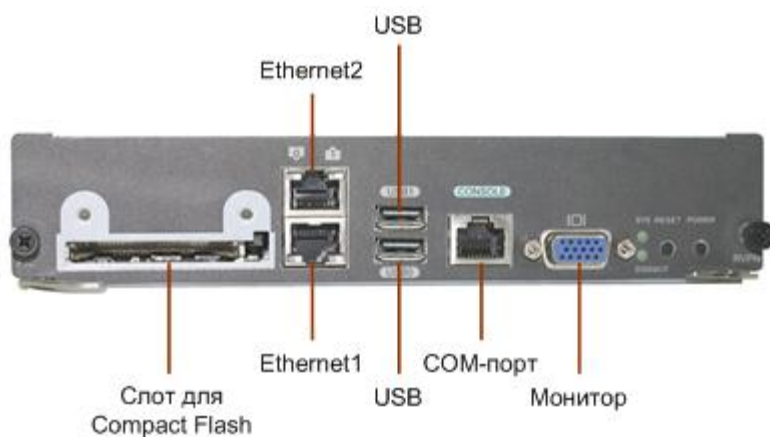


Рисунок 8: Лицевая панель модуля VPNM

На флэш-диске установлено ПО ViPNet Coordinator Linux, которое функционирует под управлением адаптированной ОС Linux. Для блокирования свободного доступа к флэш-дискету предусмотрена крышка на винтах.

В качестве консоли можно использовать следующее оборудование:

- ноутбук, подключенный к COM-порту (COM-консоль);
- монитор и клавиатуру (обычная консоль).

Перед началом эксплуатации ПАК на нем должны быть развернуты ключевые базы ViPNet. При развертывании ключевых баз используется мобильный компьютер (ноутбук), подключенный к порту Ethernet ПАК, или USB-флэш, вставленный в USB-разъем ПАК (см. «[Первоначальное развертывание ключей](#)» на стр. 42). Управление процедурой развертывания можно осуществлять с помощью подключения по Telnet или с одной из консолей.

После успешного развертывания ключевых баз ViPNet подключиться к ПАК по Telnet невозможно. Дальнейшее администрирование комплекса осуществляется локально с одной из возможных консолей и/или удаленно по протоколу SSH (см. «[Настройка ПАК с помощью командного интерпретатора](#)» на стр. 60).

Выбор консоли при загрузке ОС

Управление ПАК ViPNet Coordinator HW возможно только с одной консоли, несмотря на то, что к нему можно подключить одновременно обе консоли (обычную и СОМ-консоль, кроме ПАК HW100 G2). Поэтому при каждой загрузке ОС, независимо от количества подключенных консолей, пользователю предлагается выбрать консоль для дальнейшей работы. При этом для ПАК ViPNet Coordinator HW100 и HW1000 консолью по умолчанию является обычная консоль, а для ПАК ViPNet Coordinator HW-VPNМ – СОМ-консоль.

Выбор консоли происходит следующим образом:

- 1** На все консоли, подключенные к ПАК, в течение 10 секунд выводится приглашение нажать любую клавишу. Если в течение этого времени ни на одной из консолей не будет нажата клавиша, то считается, что клавиша нажата на консоли по умолчанию.
- 2** На консоль, на которой нажата клавиша, выводится список возможных консолей и приглашение выбрать нужную консоль. Если в течение 5 секунд консоль не будет выбрана, то автоматически выбирается консоль по умолчанию.

Все последующие сообщения о загрузке ОС и приглашение для входа в систему будут выводиться на выбранную консоль.



3

Лицензирование ViPNet Coordinator HW

Лицензионные ограничения ПАК ViPNet Coordinator HW100	39
Лицензирование ПАК ViPNet Coordinator HW1000	40
Лицензирование ПАК ViPNet Coordinator HW-VPNM	41

Лицензионные ограничения ПАК ViPNet Coordinator HW100

Функциональность ПАК ViPNet Coordinator HW100 определяется лицензией, предоставляемой компанией «ИнфоТеКС» при его поставке. Лицензия устанавливает ограничения на количество одновременно поддерживаемых туннелей и на поддержку функции Сервера-маршрутизатора. Возможность использования ПАК в качестве Сервера-маршрутизатора зависит от его аппаратной конфигурации: эта функциональность поддерживается только на ПАК расширенной конфигурации, т.е. на ПАК с дополнительным жестким диском. Дополнительный диск используется для хранения очереди конвертов MFTR. На ПАК базовой конфигурации функция Сервера-маршрутизатора не поддерживается.

Внимание! Базовая конфигурация ПАК ViPNet Coordinator HW100 имеет ограниченную функциональность:

Не поддерживается функционал шлюзового Координатора.



Вследствие этого при формировании в ЦУСе сети ViPNet сетевой узел, который будет разворачиваться на ПАК ViPNet Coordinator HW100 базовой конфигурации, нельзя регистрировать в качестве пограничного шлюзового Координатора в другие ViPNet-сети. В случае невыполнения данного требования работоспособность ПАК не гарантируется!

Лицензия определяет прикладную задачу, в которой должен быть зарегистрирован ПАК. Соответствие аппаратной конфигурации ПАК прикладной задаче, в которой он зарегистрирован, контролируется при развертывании на ПАК ключевых баз ViPNet (см. [«Первоначальное развертывание ключей»](#) на стр. 42).

Регистрация узлов в прикладных задачах и создание лицензии осуществляется в ЦУСе в процессе формирования сети ViPNet. Подробное описание действий по формированию сети ViPNet содержится в документе «ViPNet Administrator Центр управления сетью. Руководство администратора».

Лицензирование ПАК ViPNet Coordinator HW1000

Для нормального функционирования ПАК ViPNet Coordinator HW1000 необходимо наличие лицензии, в которой должна быть разрешена прикладная задача «Координатор HW1000». В этой прикладной задаче должен быть зарегистрирован сетевой узел, который будет разворачиваться на ПАК ViPNet Coordinator HW1000. Наличие регистрации проверяется при развертывании на ПАК ключевых баз ViPNet (см. «[Первоначальное развертывание ключей](#)» на стр. 42).

На базе ПАК ViPNet Coordinator HW1000 можно организовать кластер горячего резервирования, который создается путем объединения двух ПАК. При этом в сети ViPNet кластер горячего резервирования представляет собой один сетевой узел, который необходимо зарегистрировать в прикладных задачах «Координатор HW1000» и «ViPNet Failover».



Примечание. Регистрация узлов для кластера горячего резервирования должна проводиться в определенной последовательности: сначала узел следует зарегистрировать в задаче «Координатор HW1000», затем – в задаче «ViPNet Failover».

Регистрация узлов в прикладных задачах и создание лицензии осуществляется в ЦУСе в процессе формирования сети ViPNet. Подробное описание действий по формированию сети ViPNet содержится в документе «ViPNet Administrator Центр управления сетью. Руководство администратора».

Лицензирование ПАК ViPNet Coordinator HW-VPNM

Для нормального функционирования ПАК ViPNet Coordinator HW-VPNM необходимо наличие лицензии, в которой должна быть разрешена прикладная задача «Координатор HW-VPNM». В этой прикладной задаче должен быть зарегистрирован сетевой узел, который будет разворачиваться на ПАК ViPNet Coordinator HW-VPNM. Наличие регистрации проверяется при развертывании на ПАК ключевых баз ViPNet (см. [«Первоначальное развертывание ключей»](#) на стр. 42).

На базе ПАК ViPNet Coordinator HW-VPNM можно организовать кластер горячего резервирования, который создается путем объединения двух ПАК. При этом в сети ViPNet кластер горячего резервирования представляет собой один сетевой узел, который необходимо зарегистрировать в прикладных задачах «Координатор HW-VPNM» и «ViPNet Failover».



Примечание. Регистрация узлов для кластера горячего резервирования должна проводиться в определенной последовательности: сначала узел следует зарегистрировать в задаче «Координатор HW-VPNM», затем – в задаче «ViPNet Failover».

Регистрация узлов в прикладных задачах и создание лицензии осуществляется в ЦУСе в процессе формирования сети ViPNet. Подробное описание действий по формированию сети ViPNet содержится в документе «ViPNet Administrator Центр управления сетью. Руководство администратора».



4

Первоначальное развертывание ключей

О первоначальном развертывании ключевых баз ViPNet	43
Подготовка к развертыванию ключевых баз	44
Процедура развертывания ключевых баз	46

О первоначальном развертывании ключевых баз ViPNet

Перед началом эксплуатации ПАК ViPNet Coordinator HW на нем необходимо развернуть ключевые базы ViPNet. Без развертывания ключевых баз работа ПАК и управление им, осуществляемое с помощью командного интерпретатора, будут невозможны.

Развертывание ключевых баз осуществляется с помощью процедуры, которая позволяет выполнить:

- первоначальное развертывание ключевых баз после установки ПАК;
- импорт ключевых баз, справочников и настроек служб ViPNet на уже функционирующий ПАК (после обновления на нем версии ПО ViPNet или для переноса на него ключевых баз, справочников и настроек с другого действующего ПАК ViPNet Coordinator HW).

Для первоначального развертывания ключевых баз необходимо наличие файла `.dst` (дистрибутива ключевых баз), для импорта – наличие файла `.vbe`, полученного в результате экспорта ключевых баз, справочников и настроек на действующем ПАК (см. [«Экспорт и импорт ключевых баз, справочников и настроек»](#) на стр. 180).

Развертывание ключевых баз можно выполнить одним из следующих способов:

- С помощью мобильного компьютера (ноутбука), который подключается к порту Ethernet1 ПАК.
- С помощью флэш-памяти USB (USB-флэш), которая вставляется в USB-разъем ПАК.

При любом способе развертывания предполагается, что на ноутбуке или USB-флэш находится файл дистрибутива ключевых баз или файл экспорта, который в ходе процедуры развертывания переносится на ПАК.

Подготовка к развертыванию ключевых баз

Первоначальное развертывание ключевых баз с помощью ноутбука

Для развертывания ключевых баз с помощью мобильного компьютера (ноутбука) требуется следующее дополнительное оборудование и ПО:

- ноутбук с сетевой картой Ethernet и установленной ОС Windows XP или Windows Vista;
- кроссированный кабель Ethernet (сетевой кабель для непосредственного соединения компьютеров друг с другом).

При развертывании ключевых баз с помощью ноутбука используются стандартные службы Telnet и TFTP. В ОС Windows XP эти службы по умолчанию включены. В ОС Windows Vista эти службы по умолчанию отключены и их необходимо включить вручную. Для включения служб в ОС Windows Vista выполните следующее:

- 1 Выберите **Start > Control Panel > Programs and Features**.
- 2 Зайдите в меню **Turn Windows features on or off** и включите службы **TFTP Client** и **Simple TCP/IP services**.

Кроме того, на время развертывания ключевых баз на ноутбуке с ОС Windows Vista отключите следующие службы безопасности (если они включены):

- Windows Firewall;
- Windows Defender;
- Windows Update;
- отключите защиту по всем параметрам в Internet Explorer (меню **Internet Options**, закладка **Security**).

Перед началом развертывания ключевых баз выполните следующее:

- 1 Заранее перенесите на ноутбук дистрибутив ключевых баз (файл `.dst`) или файл экспорта (`.vbe`).



Примечание. Обычно файл экспорта переносится на ноутбук при выполнении процедуры экспорта (см. «[Экспорт и импорт ключевых баз, справочников и настроек](#)» на стр. 180).

- 2 Подключите ноутбук к порту Ethernet1 ПАК с помощью кросс-кабеля и установите вручную на сетевом интерфейсе ноутбука IP-адрес `169.254.241.5`.
- 3 Подключитесь к ПАК по Telnet (с помощью стандартного Telnet-клиента) по адресу `169.254.241.1` либо подключите консоль (монитор и клавиатуру).

Подготовка к развертыванию ключевых баз с помощью USB-флэш

Для развертывания ключевых баз с помощью флэш-памяти USB (USB-флэш) требуется предварительно отформатировать USB-флэш в одну из поддерживаемых файловых систем: FAT32 или ext2.

Перед началом развертывания ключевых баз выполните следующее:

- 1 Заранее перенесите на USB-флэш дистрибутив ключевых баз (файл `.dst`) или файл экспорта (`.vbe`).



Примечание. Обычно файл экспорта переносится на USB-флэш при выполнении процедуры экспорта (см. «[Экспорт и импорт ключевых баз, справочников и настроек](#)» на стр. 180).

- 2 Подключите к ПАК консоль (монитор и клавиатуру).

Процедура развертывания ключевых баз

Для начала процедуры развертывания ключевых баз введите логин (`vipnet`) и пароль (`vipnet`). После входа в систему автоматически запускается мастер, выполняющий процедуру. При использовании консоли (монитора и клавиатуры) мастер может работать в одном из двух режимов: в обычном консольном режиме или в полноэкранном режиме с эмуляцией графического интерфейса. Выбор режима работы мастера производится сразу после его запуска. При подключении к ПАК по Telnet мастер работает только в консольном режиме. В этом случае отсутствует возможность выбора режима работы, мастер принудительно устанавливает консольный режим.

При описании процедуры развертывания ключевых баз приведены оба варианта работы мастера – в консольном режиме и в полноэкранном режиме.



Внимание! В полноэкранном режиме работы мастера не поддерживаются клавиши доступа к кнопкам (так называемые «горячие клавиши»).

В полноэкранном режиме переход от одного шага процедуры к следующему происходит с помощью кнопки **Next**. Кроме того, на каждом шаге процедуры предусмотрены кнопки **Back** (кроме первого шага) и **Cancel**.

По кнопке **Back** происходит возврат на предыдущий шаг процедуры. Однако в случае каких-либо ошибок, когда требуется нажать кнопку **Back**, может происходить откат на несколько шагов назад. Все такие случаи оговариваются отдельно с указанием шага, с которого продолжится выполнение процедуры.

С помощью кнопки **Cancel** можно прервать выполнение процедуры на любом шаге. Перед тем, как прервать процедуру, мастер запрашивает подтверждение, при отказе от прерывания выполнение процедуры продолжается. В случае прерывания процедуры состояние системы не изменяется, она остается в том состоянии, в котором была до запуска процедуры.

Процедура развертывания ключевых баз выполняется в следующей последовательности:

- 1 Мастер определяет тип терминала, на котором он запущен:
 - Если это консоль (монитор и клавиатура), мастер предлагает выбрать режим работы (консольный или полноэкранный). Выберите нужный режим.

- Если это подключение по Telnet, мастер автоматически устанавливает консольный режим.
- 2 Мастер сообщает о необходимости развертывания ключевых баз и запрашивает подтверждение. Ответьте утвердительно и нажмите ввод (в полноэкранном режиме нажмите кнопку **Next**).

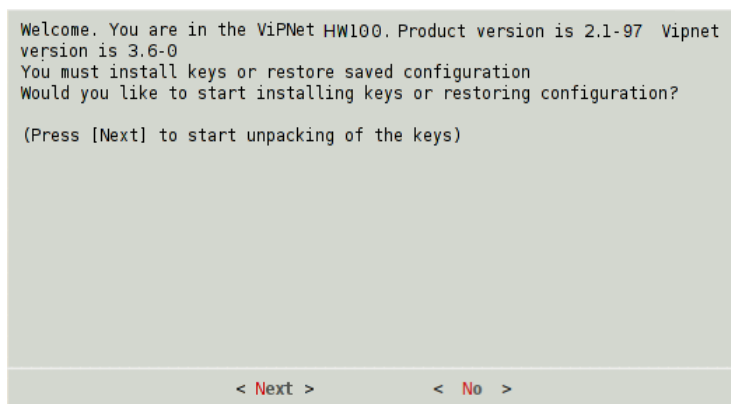


Рисунок 9: Запрос о начале развертывания ключевых баз

- 3 Мастер предлагает выбрать способ переноса файла (по TFTP или с USB-флэш). Выберите нужный способ (в полноэкранном режиме установите переключатель в нужное положение и нажмите кнопку **Next**).

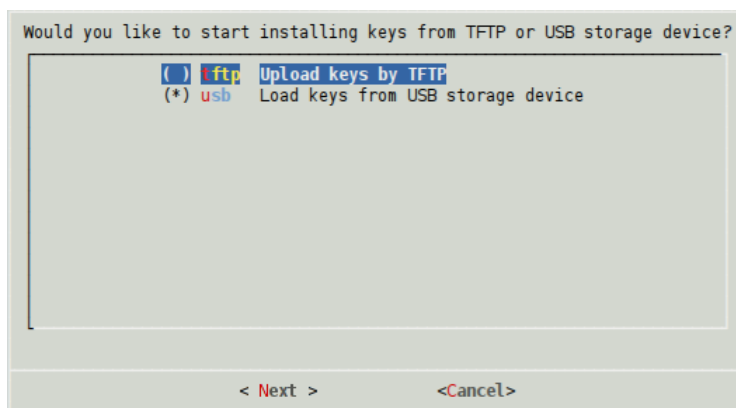


Рисунок 10: Выбор способа переноса файла на ПАК

- 4 В зависимости от выбранного способа переноса файла мастер предлагает перенести файл по TFTP или вставить USB-флэш.

Если выбран способ переноса по TFTP, перенесите на ПАК нужный файл (файлы) с помощью команды `tftp -i 169.254.241.1 put <имя файла>`, после чего нажмите ввод (в полноэкранном режиме нажмите кнопку **Next**).

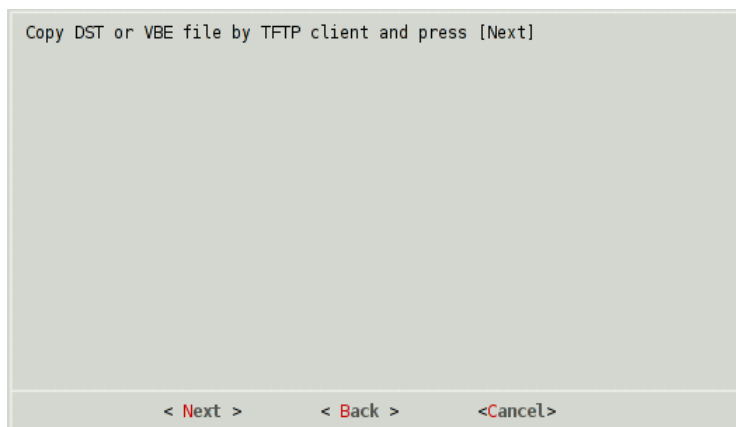


Рисунок 11: Подтверждение переноса файла по TFTP

Если выбран способ переноса с USB-флэш, вставьте USB-флэш в один из USB-разъемов ПАК, после чего нажмите ввод (в полноэкранном режиме нажмите кнопку **Next**). Мастер скопирует с USB-флэш на ПАК дистрибутивы ключевых баз и файлы экспорта (если они есть).

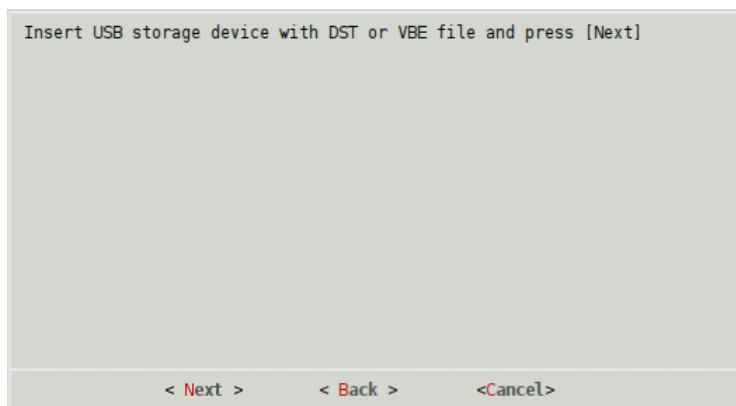


Рисунок 12: Подтверждение вставления USB-флэша

5 Мастер проверяет наличие на ПАК файлов .dst и .vbe:

- Если файлов нет, мастер сообщает об этом и предлагает заново перенести файл (возврат к шагу 3). В полноэкранном режиме нажмите в окне сообщения кнопку **Back**, произойдет возврат к предыдущему шагу 4.
- Если файлов несколько, мастер выводит пронумерованный список файлов и предлагает выбрать нужный файл. Для файлов .dst мастер дополнительно выводит имена и идентификаторы сетевых узлов, которым они соответствуют. В полноэкранном режиме список выводится также в случае, если найден только один файл.

В консольном режиме, если найдено больше 20-и файлов, мастер выводит список постранично по 20 файлов на странице. На каждой странице мастер

предлагает выбрать нужный файл либо перейти к следующей или первой странице.

В полноэкранном режиме длинные имена файлов могут быть не видны в списке полностью. Чтобы увидеть полное имя, выберите файл в списке – его имя будет показано под окном мастера.

В консольном режиме введите номер файла. Если номер не введен или введен некорректный номер, мастер сообщает об этом и предлагает заново ввести номер файла. В полноэкранном режиме выберите файл в списке и нажмите кнопку **Next**.

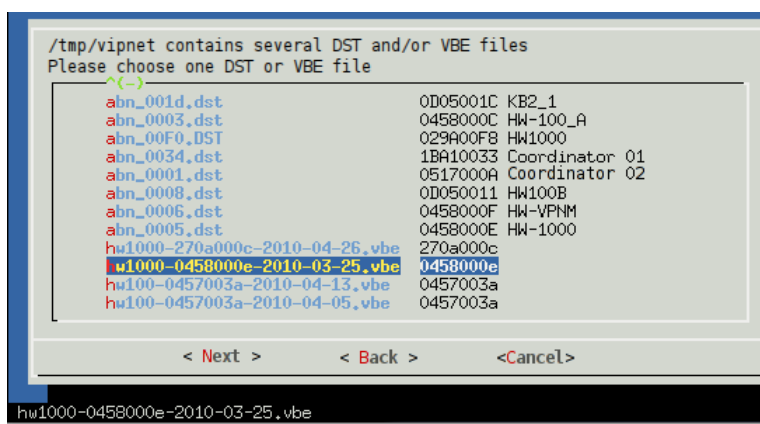


Рисунок 13: Выбор файла для развертывания ключевых баз

Если файл один, мастер автоматически выбирает его для развертывания ключевых баз (только в консольном режиме).

1 Мастер выполняет развертывание ключевых баз из выбранного файла.

Если на предыдущем шаге выбран файл дистрибутива .dst:

- Мастер пытается распаковать дистрибутив ключевых баз:
 - Если распаковать дистрибутив не удастся, мастер сообщает об этом и предлагает заново перенести файл дистрибутива (возврат к шагу 3). В полноэкранном режиме нажмите в окне сообщения кнопку **Back**, произойдет возврат к шагу 5.
 - Если дистрибутив успешно распакован, мастер запрашивает пароль доступа к этому дистрибутиву.

- Мастер предлагает ввести пароль доступа к дистрибутиву ключевых баз. Введите пароль. В полноэкранном режиме после ввода пароля нажмите кнопку **Next**.

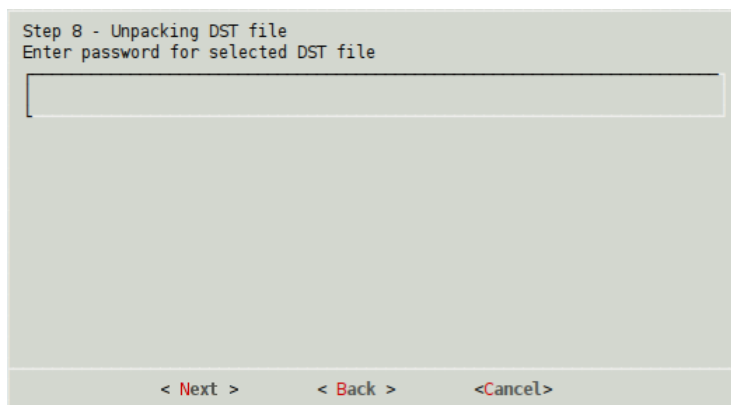


Рисунок 14: Ввод пароля доступа к дистрибутиву ключевых баз

- Мастер проверяет пароль:
 - Если введен неверный пароль, мастер сообщает об этом и предлагает заново ввести пароль либо перенести на ПАК другой дистрибутив. В зависимости от ответа происходит возврат к шагу 3 или повторный ввод пароля. В полноэкранном режиме нажмите в окне сообщения кнопку **Back**, произойдет возврат к странице ввода пароля.
 - Если пароль верен, мастер сообщает об успешном развертывании ключевых баз и переходит к следующему шагу 7.

Если на предыдущем шаге выбран файл экспорта .vbe:

- Мастер предлагает ввести пароль доступа к файлу экспорта. Введите пароль. В полноэкранном режиме после ввода пароля нажмите кнопку **Next**.

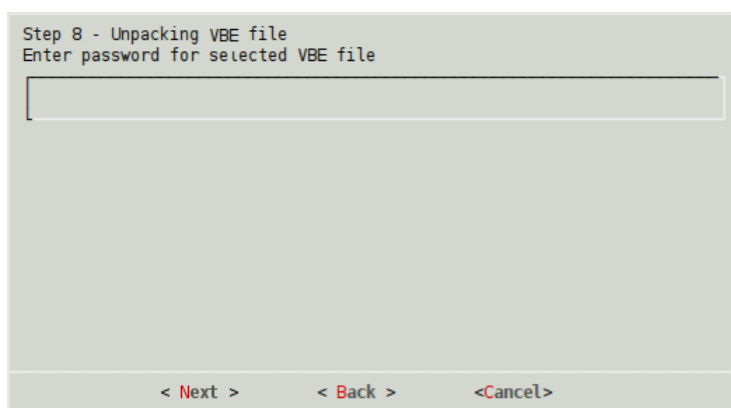


Рисунок 15: Ввод пароля доступа к файлу экспорта

- Мастер проверяет пароль:
 - Если введен неверный пароль, мастер сообщает об этом и предлагает заново ввести пароль либо перенести на ПАК другой файл экспорта. В зависимости от ответа происходит возврат к шагу 3 или повторный ввод пароля. В полноэкранном режиме нажмите в окне сообщения кнопку **Back**, произойдет возврат к странице ввода пароля.
 - Если пароль верен, мастер переходит к импорту ключевых баз.
- Мастер пытается выполнить импорт ключевых баз, справочников и настроек служб ViPNet:
 - Если выполнить импорт не удастся, мастер сообщает об этом и предлагает заново перенести файл экспорта (возврат к шагу 3). В полноэкранном режиме нажмите в окне сообщения кнопку **Back**, произойдет возврат к шагу 5.
 - Если импорт выполнен успешно, мастер сообщает об этом и переходит к шагу 18.



Примечание. Шаги 7-17 предназначены для настройки ПАК. При развертывании ключевых баз из файла .vbe эти шаги пропускаются, так как все настройки импортируются из файла экспорта. В результате успешного импорта на ПАК будут установлены те настройки, которые были на момент выполнения процедуры экспорта.



Примечание. Шаги 7-9 предназначены для конфигурирования сетевых интерфейсов и повторяются для каждого интерфейса ПАК.

- 1 Мастер спрашивает, нужно ли включить интерфейс:
 - При положительном ответе мастер переходит к следующему шагу 8. В полноэкранном режиме для включения интерфейса установите переключатель в положение **UP** и нажмите кнопку **Next**.

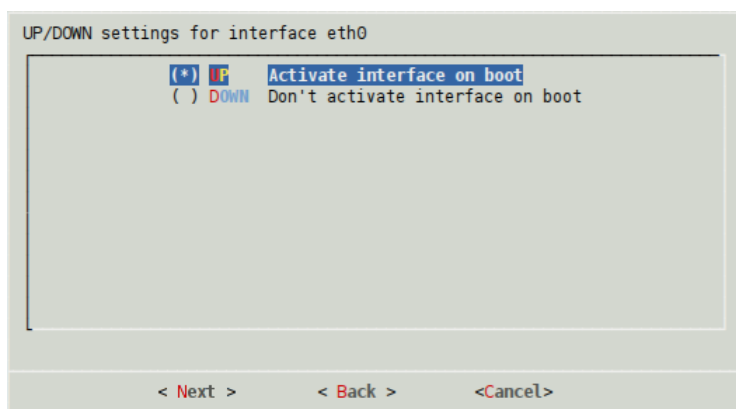


Рисунок 16: Включение или выключение интерфейса

- При отрицательном ответе мастер повторяет этот шаг для следующего интерфейса. Если на данном шаге сконфигурирован последний интерфейс, мастер переходит к шагу 10. В полноэкранном режиме для выключения интерфейса установите переключатель в положение **DOWN** и нажмите кнопку **Next**.
- 2 Мастер спрашивает, нужно ли установить для интерфейса режим DHCP:
- При положительном ответе мастер переходит к шагу 7 для следующего интерфейса. Если на данном шаге сконфигурирован последний интерфейс, мастер переходит к шагу 10. В полноэкранном режиме для включения режима DHCP установите переключатель в положение **DHCP** и нажмите кнопку **Next**.

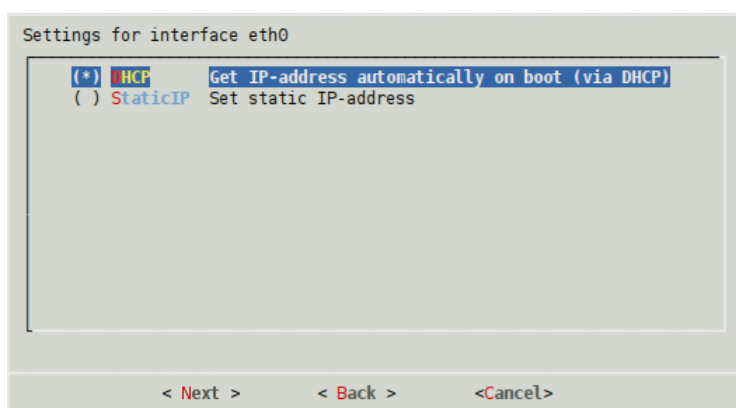


Рисунок 17: Включение/выключение режима DHCP для интерфейса

- При отрицательном ответе мастер переходит к следующему шагу 9. В полноэкранном режиме для выключения режима DHCP установите переключатель в положение **StaticIP** и нажмите кнопку **Next**.

- 3 Мастер последовательно запрашивает IP-адрес интерфейса и маску подсети. Введите IP-адрес и маску. В полноэкранном режиме введите параметры интерфейса и нажмите кнопку **Next**.



Static IP-address settings for eth0

Interface Name	eth0
IP-address	10.0.8.33
Netmask	255.255.255.0

< Next > < Back > <Cancel>

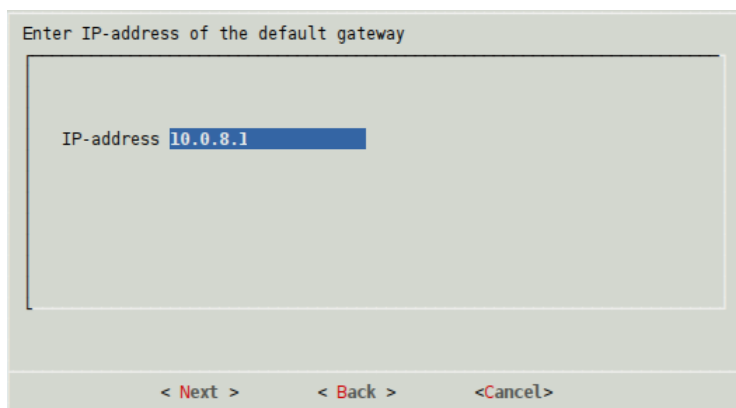
Рисунок 18: Установка параметров интерфейса



Примечание. Если параметры интерфейса были установлены ранее, то они подставляются в соответствующие поля ввода (в полноэкранном режиме).

Если сконфигурированный на данном шаге интерфейс не последний, мастер переходит к шагу 7 для следующего интерфейса, иначе переходит к следующему шагу 10.

- 4 Мастер запрашивает IP-адрес шлюза по умолчанию. Введите IP-адрес. В полноэкранном режиме после ввода адреса нажмите кнопку **Next**.



Enter IP-address of the default gateway

IP-address	10.0.8.1
------------	----------

< Next > < Back > <Cancel>

Рисунок 19: Установка IP-адреса шлюза по умолчанию



Примечание. Если адрес шлюза по умолчанию был установлен ранее, то он подставляется в поле ввода (в полноэкранном режиме).

5 Мастер сообщает, что в качестве DNS-серверов по умолчанию используются корневые DNS-серверы, и при наличии подключения к Интернету нет необходимости использовать другие серверы. Затем мастер спрашивает, нужно ли добавить адрес DNS-сервера:

- При положительном ответе мастер переходит к следующему шагу 12. В полноэкранном режиме для добавления адреса DNS-сервера установите переключатель в положение **Yes (Add custom DNS server)** и нажмите кнопку **Next**.

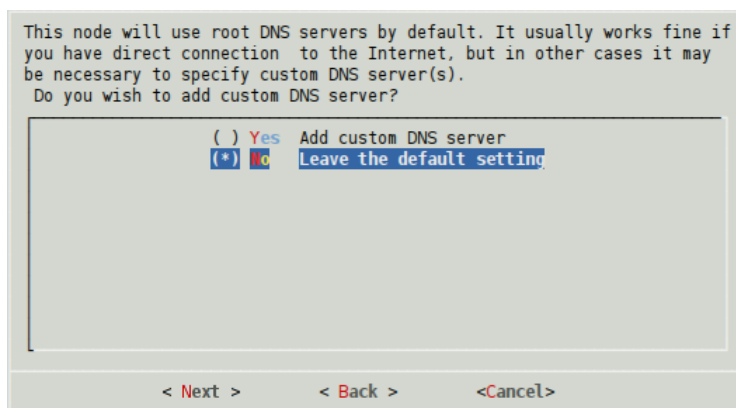


Рисунок 20: Запрос на добавление адреса DNS-сервера

- При отрицательном ответе мастер переходит к шагу 13. В полноэкранном режиме для отказа от добавления адреса DNS-сервера установите переключатель в положение **No (Leave the default setting)** и нажмите кнопку **Next**. В этом случае будут использоваться либо настройки по умолчанию, либо текущие настройки (если это не первое развертывание ключевых баз).
- 6 Мастер запрашивает IP-адрес DNS-сервера. Введите IP-адрес. В полноэкранном режиме после ввода адреса нажмите кнопку **Next**.

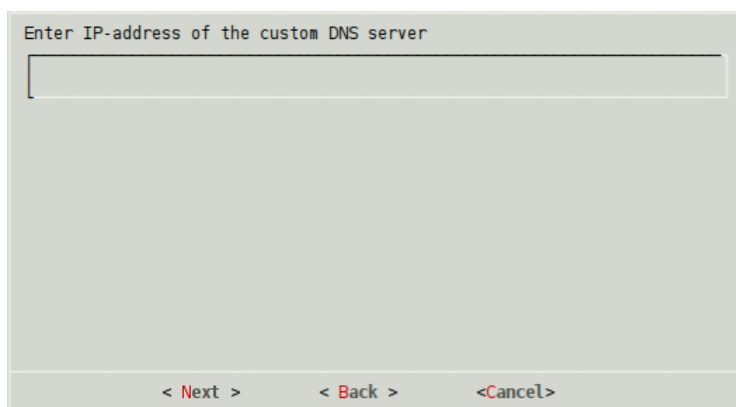


Рисунок 21: Задание IP-адреса DNS-сервера

- 7 Мастер сообщает, что для синхронизации системного времени по умолчанию используются публичные NTP-серверы точного времени. Затем мастер спрашивает, нужно ли добавить NTP-сервер:
 - При положительном ответе мастер переходит к следующему шагу 14. В полноэкранном режиме для добавления NTP-сервера установите переключатель в положение **Yes (Add custom NTP server)** и нажмите кнопку **Next**.

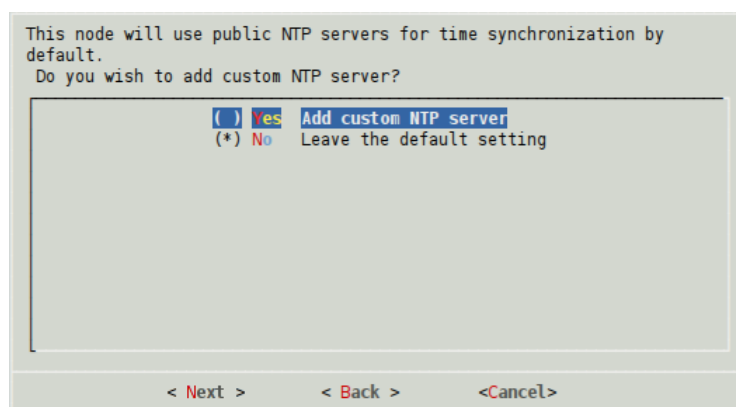


Рисунок 22: Запрос на добавление NTP-сервера

- При отрицательном ответе мастер переходит к шагу 15. В полноэкранном режиме для отказа от добавления NTP-сервера установите переключатель в положение **No (Leave the default setting)** и нажмите кнопку **Next**. В этом случае будут использоваться либо настройки по умолчанию, либо текущие настройки (если это не первое развертывание ключевых баз).
- 8 Мастер запрашивает IP-адрес NTP-сервера. Введите IP-адрес или DNS-имя. В полноэкранном режиме после ввода нажмите кнопку **Next**.

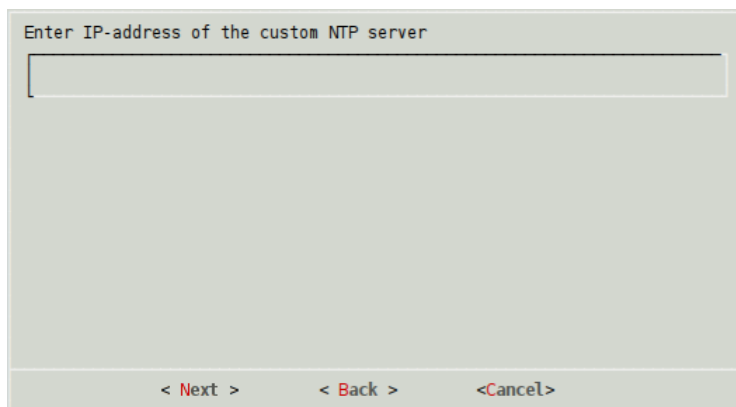


Рисунок 23: Задание IP-адреса NTP-сервера



Примечание. Шаги 15-17 предназначены для задания временной зоны (часового пояса) перед последующей установкой на ПАК текущего времени. Временная зона должна соответствовать географическому месторасположению ПАК.

При развертывании ключевых баз из файла .vbe эти шаги пропускаются, так как настройки временной зоны импортируются из файла экспорта.

- 1 Мастер выводит пронумерованный список континентов и предлагает выбрать континент. Введите номер своего континента. В полноэкранном режиме выберите континент в списке и нажмите кнопку **Next**.



Рисунок 24: Выбор континента

Если на ПАК необходимо установить время UTC, выберите в списке последний элемент. В этом случае сразу выводится информация о текущем времени UTC и запрашивается подтверждение на его установку (см. шаг 17).

- 2 Мастер выводит пронумерованный список стран, расположенных на выбранном континенте, и предлагает выбрать страну. Введите номер своей страны. В полноэкранном режиме выберите страну в списке и нажмите кнопку **Next**.

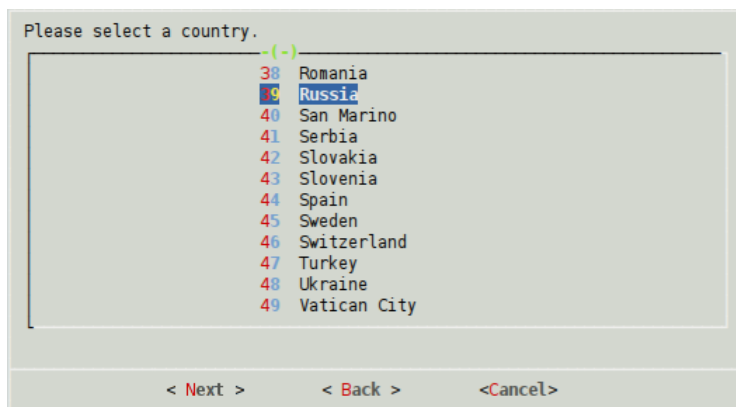


Рисунок 25: Выбор страны

- 3 Мастер выводит пронумерованный список временных зон (с указанием их общепринятых названий), имеющихся в выбранной стране, и предлагает выбрать временную зону. Введите номер своей временной зоны. В полноэкранном режиме выберите временную зону в списке и нажмите кнопку **Next**.



Рисунок 26: Выбор временной зоны

Если в выбранной на предыдущем шаге стране есть только одна временная зона, она выбирается автоматически. После выбора временной зоны выводится информация о текущем времени в этой зоне и запрашивается подтверждение на ее установку:

- При положительном ответе мастер переходит к следующему шагу 18. В полноэкранном режиме для установки выбранной временной зоны нажмите кнопку **Yes**.

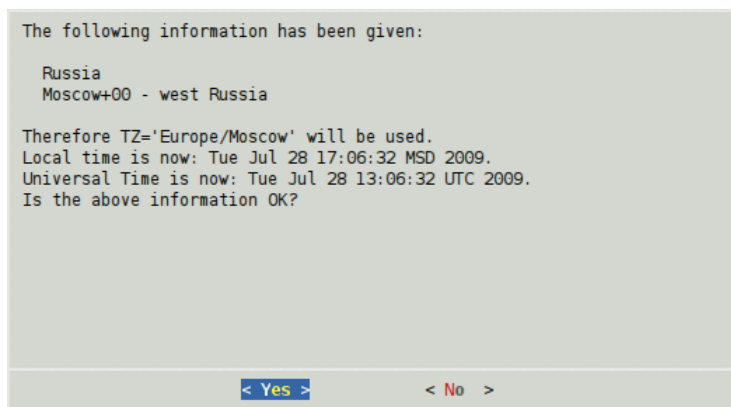


Рисунок 27: Запрос на установку временной зоны

- При отрицательном ответе мастер предлагает заново выбрать временную зону (возвращается к шагу 15). В полноэкранном режиме для выбора другой временной зоны нажмите кнопку **No**.

4 Мастер выводит текущую дату и предлагает изменить ее:

- Если дату необходимо изменить, введите нужное значение. В полноэкранном режиме установите нужную дату и нажмите кнопку **Next**.

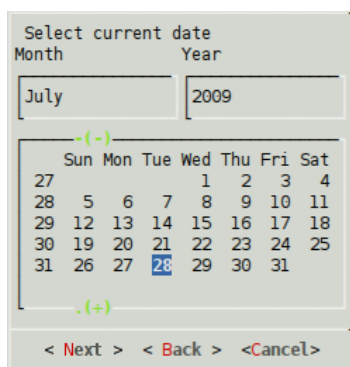


Рисунок 28: Установка текущей даты

- Если дату менять не надо, нажмите ввод. В полноэкранном режиме нажмите кнопку **Next**.

5 Мастер выводит текущее время и предлагает изменить его:

- Если время необходимо изменить, введите нужное значение. В полноэкранном режиме установите нужное время и нажмите кнопку **Next**.

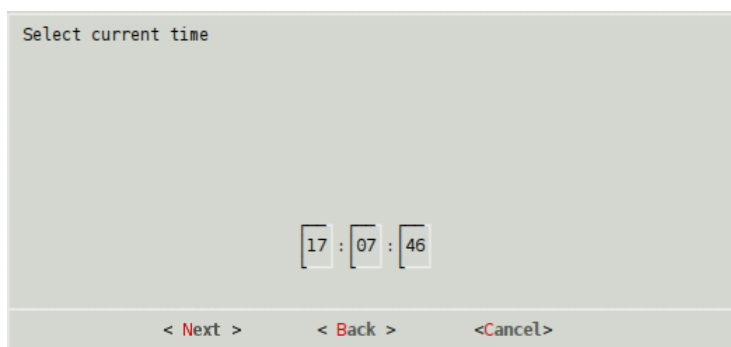


Рисунок 29: Установка текущего времени

- Если время менять не надо, нажмите ввод. В полноэкранном режиме нажмите кнопку **Next**.
- 6** Мастер сообщает об успешном завершении первоначальной настройки и спрашивает, нужно ли запустить командный интерпретатор:
- При положительном ответе запускается командный интерпретатор, мастер завершает свою работу. В полноэкранном режиме нажмите кнопку **Run Command shell**.



Рисунок 30: Сообщение об успешном завершении первоначальной настройки

- При отрицательном ответе мастер завершает свою работу без запуска командного интерпретатора. В полноэкранном режиме нажмите кнопку **Finish**.

Командный интерпретатор предусматривает возможность удаления ключей ViPNet с помощью команды `admin remove keys` (см. «Команды группы `admin`» на стр. 79). После удаления ключей можно заново выполнить развертывание ключевых баз.



5

Настройка ПАК с помощью командного интерпретатора

О командном интерпретаторе	61
Интерфейс командного интерпретатора	62
Средства для облегчения ввода команд	65
Команды интерпретатора	67

О командном интерпретаторе

Все операции по администрированию ПАК выполняются с помощью командного интерпретатора ViPNet. Администрирование ПАК возможно как локально с консоли ПАК (СОМ-консоли или обычной консоли), так и удаленно с других узлов сети ViPNet, связанных с ПАК. Для удаленного подключения к ПАК используется протокол SSH.

Допустимо одновременное подключение к ПАК с нескольких узлов, на каждом из которых запускается свой командный интерпретатор. При администрировании ПАК можно посмотреть информацию обо всех узлах, на которых запущен командный интерпретатор, с помощью команды `who` (см. «[Прочие команды](#)» на стр. 85). Кроме того, с помощью команды `admin kick` можно принудительно завершить работу командного интерпретатора на любом из узлов (см. «[Команды группы admin](#)» на стр. 79).

Интерфейс командного интерпретатора

Командный интерпретатор запускается автоматически после успешной авторизации пользователя в ОС Linux. Для авторизации необходимо ввести логин «vipnet» и пароль пользователя, который должен совпадать с паролем дистрибутива ключевых баз ViPNet. При вводе пароля на экране ничего не отображается, введенные символы пароля отредактировать нельзя.

Для удаленного подключения можно использовать любой SSH-клиент с парольным типом аутентификации. Для авторизации используются тот же логин (vipnet) и пароль, как и при локальной авторизации на ПАК.

Если введенный пароль неверен, на консоли появляется соответствующее сообщение и приглашение ввести пароль. Допускается не более 10-и попыток ввода пароля. После 10-и неудачных попыток ввода пароля ПАК перезагружается.

После успешной авторизации пользователя появляется приветственное сообщение и приглашение командного интерпретатора:

```
vipnet> _
```

Командный интерпретатор может находиться в одном из двух режимов: в режиме пользователя или в режиме администратора. После запуска командный интерпретатор переходит в режим пользователя (на это указывает знак > в подсказке интерпретатора). В режиме пользователя недоступны некоторые команды, требующие прав администратора.

Чтобы перейти в режим администратора, надо ввести команду enable и затем пароль администратора (см. «[Прочие команды](#)» на стр. 85). В случае ввода верного пароля интерпретатор перейдет в привилегированный режим (обозначается знаком # в подсказке):

```
vipnet> enable
administrator password: <пароль>
vipnet# _
```

Чтобы перейти назад в режим пользователя, надо нажать клавиши <Ctrl+D> или ввести команду exit (см. «[Прочие команды](#)» на стр. 85). Затем таким же образом можно выйти из командного интерпретатора. В случае выхода из командного интерпретатора на

консоли снова появится приглашение авторизоваться в ОС Linux. После успешной авторизации осуществляется старт командного интерпретатора.

В случае одновременной работы нескольких командных интерпретаторов (независимо от типа подключения – локального или удаленного) только один из них может находиться в режиме администратора. При попытке перейти в некотором командном интерпретаторе в режим администратора проверяется режим работы остальных запущенных интерпретаторов. Если остальные интерпретаторы находятся в режиме пользователя, то данный интерпретатор переходит в режим администратора. Если один из остальных интерпретаторов находится в режиме администратора, то появляется сообщение с информацией об узле, на котором работает этот интерпретатор, и с предложением принудительного отключения этого интерпретатора. При положительном ответе на предложение завершается работа интерпретатора, находящегося в режиме администратора, после чего данный интерпретатор переходит в режим администратора.

Командный интерпретатор принимает от пользователя текстовые команды, состоящие из слов, разделенных пробелами. Список доступных групп команд можно посмотреть, введя символ «?»:

```
vipnet> ?
inet          manage routing and interfaces
failover      manage the failover daemon
iplir         manage the iplir daemon
mftp          manage the MFTP daemon
enable        go to administrator mode
exit          exit the interpreter
version       show versions of a product and its components
who           show vipnet sessions
machine       halt or reboot the machine
debug
vipnet> _
```

В левой колонке выводится первое слово группы команд, в правой колонке – краткое пояснение ее назначения.

Символ «?» можно использовать также в процессе ввода команды. В этом случае командный интерпретатор предложит варианты завершения текущего или следующего слова команды, в зависимости от положения курсора:

```
vipnet> machi?
machine halt or reboot the machine
vipnet> machi_
```

```
vipnet> machine ?
halt          switch the machine off
reboot       reboot the machine
show         display statistics
vipnet> machine _
```

Если слово, на котором находится курсор, является началом только одной команды интерпретатора, то можно автоматически дополнить слово до команды, нажав клавишу **<Tab>**:

```
vipnet> machi<Tab>
vipnet> machine _
```

Командный интерпретатор запоминает команды, введенные в течение сеанса работы. С помощью стрелок вверх и вниз можно листать историю команд:

```
vipnet> machine show date
Mon Aug 10 19:55:42 MSD 2009
vipnet> machine show uptime
19:55:48 up 9:57, 3 users, load average: 1.13, 1.17, 1.11
vipnet> <стрелка вверх>
vipnet> machine show uptime <стрелка вверх>
vipnet> machine show date
```

Текущую команду можно редактировать обычным образом (стирать символы клавишами **<Backspace>** и **<Delete>**, перемещаться по тексту с помощью стрелок влево и вправо).

Командный интерпретатор поддерживает традиционные для Unix сочетания клавиш:

<Ctrl+U>	стереть всю команду
<Ctrl+K>	стереть все от курсора до конца строки
<Ctrl+A> и <Ctrl+E>	перейти в начало и конец строки соответственно
<Ctrl+B> и <Ctrl+F>	перейти на один символ назад и вперед соответственно
<Ctrl+H>	стереть символ перед курсором
<Ctrl+W>	стереть слово перед курсором

Средства для облегчения ввода команд

Для облегчения ввода длинных команд предусмотрены следующие средства.

Сокращенный ввод команд

Слова команды распознаются по минимальному числу символов, необходимых для того, чтобы отличить ее от других команд, ввод которых возможен в текущем месте.

Например: если существуют только две команды, первое слово которых начинается с символа «i» (`inet` и `iplir`), то для указания первой команды достаточно набрать `in`, для указания второй – `ip`. Далее, если после слова `iplir` в качестве второго слова возможны только `stop` и `start`, то для их указания достаточно набрать `sto` и `sta` соответственно. Таким образом, вместо полной команды `iplir start` можно набрать `ip sta`.

Если среди введенных символов есть ошибочные символы, то вся команда считается ошибочной, даже если ее можно однозначно распознать по первым правильно введенным символам. Например, в приведенном выше примере ввод команды `iplor sta` будет ошибочным.

Автозаполнение

Ввод символа табуляции (осуществляется нажатием клавиши **<Tab>**) позволяет автоматически заполнить часть команды, если введенной к этому моменту информации достаточно для того, чтобы распознать слово команды или ее параметр.

Автозаполнение работает по месту расположения курсора и влияет только на одно слово – то, на котором стоит курсор. Автозаполнение может работать как на словах команды, так и на некоторых параметрах, если только значения параметров не вводятся пользователем произвольно (например, названия сетевых интерфейсов). Далее при описании команд явно указывается, в каких случаях при вводе параметров должно работать автозаполнение.

Контекстная подсказка

Контекстная подсказка вызывается вводом знака вопроса. Принципы ее работы во многом схожи с принципами работы автозаполнения и дополняют его.

Действие подсказки состоит в том, чтобы, исходя из положения курсора, показать пользователю его возможные действия по дальнейшему вводу команды. В отличие от автозаполнения, подсказка всегда показывает какое-либо пояснение и не изменяет введенную пользователем строку.

При вызове подсказки введенная пользователем строка (вместе с приглашением) печатается на экране, ниже печатаются пояснения подсказки, затем появляется та же строка (вместе с приглашением) для дальнейшего редактирования.

Команды интерпретатора

Команды, доступные только в режиме администратора, выделены красным цветом.

Параметры команд указаны в угловых скобках, необязательные параметры заключены в квадратные скобки.

Команды группы `inet`

Все команды данной группы, изменяющие какие-либо параметры, сохраняют значения в служебных файлах конфигурации для восстановления после перезагрузки ОС.

- `inet show interface [<интерфейс или псевдоустройство>]` – просмотр адреса и маски подсети интерфейса (или дополнительного адреса интерфейса – см. команду `inet ifconfig <интерфейс> address add`), а также состояния интерфейса (включен/выключен).

Если параметр не указан, то выводятся адреса, маски и состояния всех интерфейсов, а также все существующие дополнительные адреса интерфейсов. При вводе интерфейса работают автозаполнение и подсказка, данные для подсказки берутся из списка интерфейсов в системе.

Если для интерфейса был установлен режим DHCP (командой `inet ifconfig <интерфейс> dhcp`), то дополнительно выводится информация об этом.

- `inet show dhcp` – просмотр текущих настроек DHCP-сервера, а также его состояния (запущен/остановлен).

Настройки DHCP-сервера включают в себя признак автоматического запуска DHCP-сервера при старте ПАК и ряд параметров, необходимых для его работы (см. «Использование ПАК в качестве DHCP-сервера» на стр. 189).



Примечание. Команда `inet show dhcp` доступна только при работе ПАК в одиночном режиме (см. «Система защиты от сбоев» на стр. 169).

- `inet show dns` – просмотр текущих настроек DNS-сервера, а также его состояния (запущен/остановлен).

Настройки DNS-сервера включают в себя признак автоматического запуска DNS-сервера при старте ПАК и список адресов форвардных (forwarder) DNS-серверов

(см. «Использование ПАК в качестве DNS-сервера» на стр. 191). Список адресов форвардных DNS-серверов можно просмотреть отдельно по команде `inet dns list` (см. «Команды подгруппы `inet dns`» на стр. 74).

- `inet show ntp` – просмотр текущих настроек NTP-сервера, его состояния (запущен/остановлен), а также параметров функционирования NTP-серверов, с которыми он взаимодействует. Информация о функционировании NTP-серверов выводится только в случае, если на ПАК запущен NTP-сервер.

Настройки NTP-сервера включают в себя признак автоматического запуска NTP-сервера при старте ПАК и список NTP-серверов, используемых для синхронизации (см. «Использование ПАК в качестве NTP-сервера» на стр. 193). Список адресов NTP-серверов можно просмотреть отдельно по команде `inet ntp list` (см. «Команды подгруппы `inet ntp`» на стр. 75).

Информация о функционировании используемых NTP-серверов (заданных в файле конфигурации локального NTP-сервера) выводится в виде таблицы аналогично выводу команды `ntpq -pn`. Каждая запись таблицы относится к одному из серверов и содержит следующие параметры:

- символ, указывающий на результат отбора данного сервера в процессе выбора претендентов (кандидатов) на роль источника синхронизации, может принимать следующие значения:
 - пробел – сервер исключен из списка кандидатов, т.к. имеет слишком высокий уровень (`stratum`) и/или не может быть проверен;
 - 'x' – сервер исключен из списка кандидатов, т.к. использует некорректный алгоритм;
 - '.' – сервер выбран из конца списка кандидатов;
 - '-' – сервер исключен из списка кандидатов алгоритмом кластеризации;
 - '+' – сервер включен в конечный список кандидатов;
 - '#' – сервер выбран для синхронизации, однако расстояние между ним и локальным сервером превышает допустимый максимум;
 - '*' – сервер является текущим источником синхронизации;
 - 'o' – сервер выбран для синхронизации, используется сигнал PPS.
- `remote` – IP-адрес сервера;
- `refid` – источник получения времени данным сервером (IP-адрес или имя другого сервера, GPS, PPS и т.п.);
- `st` – уровень сервера (`stratum`);
- `t` – тип сервера (`local`, `unicast`, `multicast` или `broadcast`);

- `when` – время, прошедшее с момента последнего ответа сервера (в секундах), или прочерк (-), если сервер еще ни разу не ответил;
 - `poll` – период опроса сервера (в секундах);
 - `reach` – состояние восьми последних попыток запроса времени у сервера в восьмеричном представлении (в случае успешной попытки устанавливается соответствующий бит); значение 377 означает, что все 8 последних попыток были удачными;
 - `delay` – вычисленная задержка ответов от сервера (в миллисекундах);
 - `offset` – разница во времени между локальным и удаленным сервером (в миллисекундах); чем меньше значение, тем точнее время;
 - `jitter` – дисперсия отклонения удаленных часов относительно локальных, вычисленная по нескольким последним запросам (в миллисекундах); чем меньше значение, тем точнее синхронизация.
- `inet show routing` – просмотр таблицы маршрутизации.
 - `inet show mac-address-table` – просмотр таблицы ARP.
 - `inet clear mac-address-table` – очистка таблицы ARP.

Перед очисткой запрашивается подтверждение на выполнение команды. Очистка таблицы ARP производится только при получении подтверждения.

- `inet ifconfig <интерфейс> address <адрес> netmask <маска>` – установка параметров интерфейса.

При вводе интерфейса работают автозаполнение и подсказка, данные для подсказки берутся из списка интерфейсов в системе. При вводе адреса и маски автозаполнение не работает, подсказка предлагает ввести соответственно IP-адрес и маску.

Эта команда немедленно изменяет адрес интерфейса и маску подсети в системе.

Если интерфейс используется локальным DHCP-сервером, который запущен, то появляется предупреждение о необходимости остановить DHCP-сервер перед изменением параметров интерфейса, команда не выполняется. В случае если DHCP-сервер остановлен:

- если автоматический запуск DHCP-сервера включен, то производимые командой изменения проверяются на соблюдение ограничений (см. «[Использование ПАК в качестве DHCP-сервера](#)» на стр. 189); при несоблюдении ограничений выдается сообщение об ошибке и команда не выполняется;
- если автоматический запуск DHCP-сервера выключен, команда немедленно изменяет адрес интерфейса и маску подсети в системе.



Внимание! При изменении параметров интерфейса данной командой из таблицы маршрутизации автоматически удаляются все маршруты, связанные с этим интерфейсом – маршрут для шлюза по умолчанию (default gateway) и статические маршруты! Для корректной работы ПАК в этом случае необходимо добавить маршрут по умолчанию и статические маршруты вручную с помощью команды `inet route add`.



Примечание. Если до выполнения данной команды на интерфейсе был установлен режим DHCP, то появляется предупреждение о том, что в результате будет потеряна информация о DNS- и NTP-серверах, полученная от DHCP-сервера, и что следует проверить функционирование соответствующих служб вручную.

- `inet ifconfig <интерфейс> dhcp` – установка на интерфейсе режима DHCP.

Если на интерфейсе имеются дополнительные адреса, то перед выполнением команды выдается предупреждение о том, что дополнительные адреса будут потеряны, и запрашивается подтверждение на установку режима DHCP. В результате выполнения команды для установки параметров интерфейса будет использоваться служба DHCP.



Примечание. При использовании службы DHCP в таблицу маршрутизации автоматически добавляется маршрут по умолчанию с адресом шлюза, полученным от DHCP-сервера.

- `inet ifconfig <интерфейс> address add <адрес> netmask <маска>` – добавление IP-адреса на интерфейс (см. «[Пример использования дополнительных IP-адресов на интерфейсе](#)» на стр. 244).

При вводе интерфейса работают автозаполнение и подсказка, данные для подсказки берутся из списка интерфейсов в системе. При вводе адреса и маски автозаполнение не работает, подсказка предлагает ввести соответственно IP-адрес и маску.

Если интерфейс выключен или для него установлен режим DHCP, то появляется сообщение об ошибке, команда не выполняется.

Если интерфейс имеет статический адрес, то создается псевдоустройство с заданными в команде адресом и маской. Имя псевдоустройства формируется как `<интерфейс>: <номер>`, где номер – очередной свободный номер, например `eth0:0`, `eth0:1` (дополнительные адреса нумеруются, начиная с 0).

Дополнительный адрес отображается в списке IP-адресов соответствующего интерфейса (см. «Секция [channel]» на стр. 151).

- `inet ifconfig <интерфейс> address delete <адрес> netmask <маска>` – удаление дополнительного IP-адреса с интерфейса. Если интерфейс выключен или для него установлен режим DHCP, то появляется сообщение об ошибке, команда не выполняется.

Если основной адрес интерфейса статический, то проверяется наличие заданного в команде дополнительного адреса. Если такой адрес есть, он удаляется из системы, иначе появляется сообщение об ошибке, команда не выполняется.

- `inet ifconfig <интерфейс> speed <10 | 100 | 1000 | auto> duplex <half | full>` – установка параметров скорости интерфейса.

При вводе интерфейса работают автозаполнение и подсказка, данные для подсказки берутся из списка интерфейсов в системе и исходя из возможных допустимых значений параметров. Возможные значения для параметра `speed`: 10, 100, 1000 (Мбит/с) или `auto` (по умолчанию). Возможные значения для параметра `duplex`: `half` или `full`.

При указании в качестве значения `auto` параметры скорости для соответствующего интерфейса определяются автоматически, независимо от ранее установленных значений. При этом указание параметра `duplex` является ошибочным. При указании значений, отличных от `auto`, режим автоматического определения параметров скорости интерфейса отключается. При этом указание параметра `duplex` является обязательным. Примеры задания параметров скорости:

```
inet ifconfig eth0 speed 100 duplex full
inet ifconfig eth0 speed auto
```

Установка параметров скорости интерфейса может использоваться для согласования работы внешнего интерфейса ПАК и внешнего коммутационного оборудования, подключенного к данному интерфейсу, в тех случаях, когда их автоматическое определение обрабатывает некорректно. Однако в большинстве случаев нет необходимости использовать ручную установку параметров скорости, так как они определяются автоматически.



Внимание! Неосознанное использование данной команды может быть опасно. Установка некорректных значений параметров скорости сетевого интерфейса может привести к его неработоспособности! Используйте данную команду только в том случае, если это действительно является необходимым.

- `inet ifconfig <интерфейс> up` – включение интерфейса, если он был выключен.

Если интерфейс был включен, то появляется соответствующее сообщение, состояние интерфейса не изменяется.

- `inet ifconfig <интерфейс> down` – выключение интерфейса, если он был включен.

Если интерфейс был выключен, то появляется соответствующее сообщение, состояние интерфейса не изменяется.

Если интерфейс используется локальным DHCP-сервером, который запущен, то появляется предупреждение о необходимости остановить DHCP-сервер перед выключением интерфейса, команда не выполняется. В случае если DHCP-сервер остановлен:

- если автоматический запуск DHCP-сервера включен, то команда не выполняется;
- если автоматический запуск DHCP-сервера выключен, команда выполняется.

- `inet route add <адрес назначения> gw <адрес шлюза> [netmask <маска>]` – добавление маршрута.

При вводе всех параметров автозаполнение не работает, подсказка предлагает ввести соответствующие значения. Если маска не указана, то она принимает следующие значения:

- 0.0.0.0, если адрес назначения равен 0.0.0.0;
- 255.255.255.255 в остальных случаях.

Вместо адреса назначения можно указать слово `default`, которое интерпретируется как значение 0.0.0.0 (маршрут по умолчанию). Можно добавить только один маршрут по умолчанию. При попытке добавить второй маршрут по умолчанию появляется соответствующее сообщение и маршрут не добавляется.

После добавления маршрута появляется сообщение о том, какой именно маршрут добавлен (включая маску).

- `inet route delete <адрес назначения> [netmask <маска>]` – удаление маршрута.

При вводе адреса назначения и маски автозаполнение не работает, подсказка предлагает ввести соответственно адрес и маску. Если маска не указана, то ищутся маршруты для указанного адреса назначения:

- если найден один маршрут, то он удаляется независимо от того, какая у него маска;
- если найдено несколько маршрутов с разными масками, то появляется сообщение о том, что необходимо указать маску.

Если указаны адрес назначения и маска, но в таблице маршрутизации для данного адреса назначения указана другая маска, то считается, что заданного в команде маршрута не существует, и появляется сообщение об ошибке.

- `inet ping <адрес>` – посылка эхо-запросов (ICMP-сообщений) на заданный адрес.
При вводе адреса автозаполнение не работает, подсказка предлагает ввести адрес.

Команды подгруппы `inet dhcp`

В группу команд `inet` входит ряд команд, предназначенных для настройки и управления DHCP-сервером, установленным на ПАК. Эти команды составляют подгруппу, которая начинается словами `inet dhcp`. Перед выполнением команд, изменяющих настройки DHCP-сервера (кроме настройки автоматического запуска), необходимо остановить DHCP-сервер. Производимые командами изменения проверяются на корректность и могут применяться или не применяться. Подробнее об условиях изменения настроек DHCP-сервера и требованиях к его настройкам см. раздел [Использование ПАК в качестве DHCP-сервера](#) (на стр. 189).



Примечание. Все команды подгруппы `inet dhcp` доступны только при работе ПАК в одиночном режиме (см. «[Система защиты от сбоев](#)» на стр. 169).

- `inet dhcp mode <on | off>` – включение (`on`) или выключение (`off`) автоматического запуска DHCP-сервера при старте ПАК.

По этой команде изменяется только настройка автоматического запуска DHCP-сервера, текущее состояние DHCP-сервера не изменяется.

При выполнении команды включения автоматического запуска (`on`) проверяется корректность текущих настроек DHCP-сервера. При некорректности настроек появляется детализированное сообщение об ошибках, команда не выполняется.

- `inet dhcp start` – запуск DHCP-сервера.

Перед запуском DHCP-сервера проверяется корректность текущих настроек DHCP-сервера. Если настройки некорректны, появляется детализированное сообщение об ошибках и команда не выполняется.

- `inet dhcp stop` – остановка DHCP-сервера.

- `inet dhcp interface <интерфейс>` – задание интерфейса, на котором должен работать DHCP-сервер.

При вводе интерфейса работают автозаполнение и подсказка, данные для подсказки берутся из списка интерфейсов в системе.

Перед выполнением команды проверяется корректность текущих параметров интерфейса (см. «[Использование ПАК в качестве DHCP-сервера](#)» на стр. 189).

- `inet dhcp range <начальный адрес> <конечный адрес>` – задание диапазона IP-адресов, доступных для назначения DHCP-клиентам.
Конечный адрес должен быть не меньше начального, иначе появляется сообщение об ошибке и команда не выполняется. Перед выполнением команды проверяется корректность заданного диапазона (см. «[Использование ПАК в качестве DHCP-сервера](#)» на стр. 189).
- `inet dhcp lease <время аренды>` – задание времени аренды (лизинга) IP-адресов, выделяемых DHCP-сервером клиентам (в секундах).
- `inet dhcp router <адрес>` – задание IP-адреса шлюза по умолчанию.
Перед выполнением команды проверяется корректность заданного адреса (см. «[Использование ПАК в качестве DHCP-сервера](#)» на стр. 189).
- `inet dhcp add wins <адрес>` – добавление IP-адреса WINS-сервера в файл конфигурации DHCP-сервера.
- `inet dhcp delete wins <адрес>` – удаление IP-адреса WINS -сервера.
Если заданный адрес отсутствует в файле конфигурации DHCP-сервера, то появляется предупреждение и команда не выполняется.

Команды подгруппы `inet dns`

В группу команд `inet` входит ряд команд, предназначенных для настройки и управления DNS-сервером, установленным на ПАК. Эти команды составляют подгруппу, которая начинается словами `inet dns`. Подробнее о функционировании DNS-сервера на ПАК см. [Использование ПАК в качестве DNS-сервера](#) (на стр. 191).

- `inet dns mode <on | off>` – включение (`on`) или выключение (`off`) автоматического запуска DNS-сервера при старте ПАК.
По этой команде изменяется только настройка автоматического запуска DNS-сервера, текущее состояние DNS-сервера не изменяется.
Перед выполнением команды проверяется текущее состояние DNS-сервера. Если DNS-сервер остановлен, то при выполнении команды включения автоматического запуска (`on`) появляется предупреждение о том, что для запуска DNS-сервера необходимо перезагрузить ПАК или запустить DNS-сервер вручную. Если DNS-сервер запущен, то при выполнении команды выключения автоматического запуска (`off`) появляется предупреждение о том, что для выключения автоматического запуска необходимо сначала остановить DNS-сервер вручную.
- `inet dns start` – запуск DNS-сервера.
- `inet dns stop` – остановка DNS-сервера.

- `inet dns list` – просмотр списка IP-адресов форвардных (forwarder) DNS-серверов, заданных в файле конфигурации локального DNS-сервера.

Если адреса форвардных DNS-серверов не заданы, то появляется предупреждение о том, что используются только корневые DNS-серверы.

Эту команду рекомендуется использовать для проверки существующего списка форвардных DNS-серверов перед выполнением команд добавления и удаления DNS-серверов.

- `inet dns add <адрес>` – добавление IP-адреса форвардного DNS-сервера в файл конфигурации локального DNS-сервера.

Перед выполнением команды проверяется способ получения адресов форвардных DNS-серверов. Если адреса были получены от DHCP-сервера, то появляется соответствующее предупреждение и команда не выполняется.

- `inet dns delete <адрес>` – удаление IP-адреса форвардного DNS-сервера из файла конфигурации локального DNS-сервера.

Перед выполнением команды проверяется способ получения адресов форвардных DNS-серверов. Если адреса были получены от DHCP-сервера, то появляется соответствующее предупреждение и команда не выполняется.

Если список адресов форвардных DNS-серверов был сформирован вручную с помощью команд, то при вводе адреса работают автозаполнение и подсказка, данные для подсказки берутся из файла конфигурации локального DNS-сервера.

Команды подгруппы `inet ntp`

В группу команд `inet` входит ряд команд, предназначенных для настройки и управления NTP-сервером, установленным на ПАК. Эти команды составляют подгруппу, которая начинается словами `inet ntp`. Подробнее о функционировании NTP-сервера на ПАК см. [Использование ПАК в качестве NTP-сервера](#) (на стр. 193).

- `inet ntp mode <on | off>` – включение (`on`) или выключение (`off`) автоматического запуска NTP-сервера при старте ПАК.

По этой команде изменяется только настройка автоматического запуска NTP-сервера, текущее состояние NTP-сервера не изменяется.

Перед выполнением команды проверяется текущее состояние NTP-сервера. Если NTP-сервер остановлен, то при выполнении команды включения автоматического запуска (`on`) появляется предупреждение о том, что для запуска NTP-сервера необходимо перезагрузить ПАК или запустить NTP-сервер вручную. Если NTP-сервер запущен, то при выполнении команды выключения автоматического запуска (`off`) появляется предупреждение о том, что для выключения автоматического запуска необходимо сначала остановить NTP-сервер вручную.

- `inet ntp start` – запуск NTP-сервера.
- `inet ntp stop` – остановка NTP-сервера.
- `inet ntp list` – просмотр списка IP-адресов NTP-серверов, заданных в файле конфигурации локального NTP-сервера.

Эту команду рекомендуется использовать для проверки существующего списка NTP-серверов перед выполнением команд добавления и удаления NTP-серверов.

- `inet ntp add <адрес | DNS-имя>` – добавление NTP-сервера в файл конфигурации локального NTP-сервера.

Перед выполнением команды проверяется способ получения адресов используемых NTP-серверов. Если адреса были получены от DHCP-сервера, то появляется соответствующее предупреждение и команда не выполняется.

- `inet ntp delete <адрес | DNS-имя>` – удаление публичного или регионального NTP-сервера из файла конфигурации локального NTP-сервера.

Перед выполнением команды проверяется способ получения адресов используемых NTP-серверов. Если адреса были получены от DHCP-сервера, то появляется соответствующее предупреждение и команда не выполняется.

Если список используемых NTP-серверов был сформирован вручную с помощью команд, то при вводе адреса или DNS-имени работают автозаполнение и подсказка, данные для подсказки берутся из файла конфигурации локального NTP-сервера.

Все команды группы `inet`, изменяющие какие-либо параметры, сохраняют значения в служебных файлах конфигурации для восстановления после перезагрузки ОС.

Команды группы `iplir`

- `iplir start` – запуск демона `iplir`.
- `iplir stop` – остановка демона `iplir`.
- `iplir info [<интерфейс>]` – просмотр информации о своем узле.

Перед выполнением этой команды проверяется, запущен ли демон `iplir`, и если это не так, то выдается ошибка. Если задан интерфейс, то выводится информация о своем узле и статистика ViPNet по этому интерфейсу, если не задан – только информация о своем узле. При вводе интерфейса работают автозаполнение и подсказка.

- `iplir config [<интерфейс> | firewall | sga]` – редактирование конфигурации демона `iplir`.

Перед выполнением этой команды проверяется, остановлен ли демон `iplir`. При указании служебного параметра `sga` дополнительно проверяется, остановлены ли демоны `mftpd` и `failoverd`. Если демон (демоны) не остановлен, то появляется сообщение об ошибке. Затем запускается текстовый редактор и в него загружается либо файл `iplir.conf`, если интерфейс не задан, либо файл `iplir.conf-<интерфейс>`, если интерфейс задан (см. «Общие принципы настройки» на стр. 88). При вводе интерфейса работают автозаполнение и подсказка, данные для подсказки берутся из списка интерфейсов в секции `[adapter]` файла `iplir.conf`.

Если вместо необязательного параметра `<интерфейс>` указан служебный параметр `firewall`, то вызывается на редактирование файл `firewall.conf`.

Если в команде указан служебный параметр `sga`, то вызывается на редактирование файл `sga.conf` (см. «Удаленный мониторинг и управление ПАК» на стр. 165). Если такого файла нет, то появляется сообщение о том, что на данном ПАК удаленное управление с помощью апплета SGA не поддерживается.

При сохранении отредактированного файла происходит проверка его корректности и в случае ошибки предлагается отказаться от изменений или продолжить редактирование. Если проверка прошла успешно, файл применяется для работы демона `iplir`, а информация об изменениях конфигурации сохраняется в журнале событий.

- `iplir view` – просмотр журнала пакетов.
- `iplir ping <идентификатор>` – проверка соединения с сетевым узлом.

При вводе идентификатора работают автозаполнение и подсказка, данные для подсказки берутся из списка связей. При этом подсказка отображает идентификатор сетевого узла и справа от него – название сетевого узла.

- `iplir show config [<интерфейс> | firewall | sga]` – просмотр основного файла конфигурации `iplir.conf`, либо файла конфигурации интерфейса `iplir.conf-<интерфейс>`, либо файла конфигурации файрвола `firewall.conf`, либо файла конфигурации доступа к апплету удаленного управления `sga.conf`. При выполнении этой команды не требуется остановка демонов. Если при попытке просмотра файла `sga.conf` оказывается, что такого файла нет, появляется сообщение о том, что на данном ПАК удаленное управление с помощью апплета SGA не поддерживается.



Примечание. Для завершения просмотра файла конфигурации нажмите клавишу «q».

- `iplir show thread-count` – просмотр текущего числа потоков в драйвере `iplir`.
- `iplir set thread-count <число потоков>` – установка числа потоков в драйвере `iplir`.

Возможные значения параметра – число в диапазоне от 1 до количества логических процессоров в системе. Если параметр не является числом, то появляется сообщение об ошибке. Если значение параметра выходит за границы допустимого диапазона, то число потоков устанавливается равным ближайшей границе (1 или числу процессоров). После выполнения команды на консоль выводится информация о фактически установленном числе потоков. По умолчанию число потоков в драйвере равно 1.

Команды группы `failover`

- `failover start` – запуск демона `failoverd`.
- `failover stop` – остановка демона `failoverd`.
- `failover show info` – просмотр текущей информации о состоянии системы защиты от сбоев.

Выводится следующая информация: версия продукта ПАК ViPNet Coordinator HW, версия демона `failoverd`, идентификатор и имя ПАК (как узла сети ViPNet), режим работы системы защиты от сбоев, локальное время на узле, текущая информация о состоянии управляющего демона, демонов `mftpd` и `failoverd`.

- `failover show config` – просмотр файла конфигурации системы защиты от сбоев.



Примечание. Для завершения просмотра файла конфигурации нажмите клавишу «q».

- `failover config edit` – редактирование конфигурации системы защиты от сбоев.
Запускается текстовый редактор и в него загружается файл конфигурации системы защиты от сбоев. При сохранении файла конфигурации проверяется, был ли он изменен. Если файл изменен, то появляется сообщение о том, что изменения вступят в силу только после перезапуска демона `failoverd`.

Приведенные команды группы `failover` можно выполнять в любом из режимов работы системы защиты от сбоев – в одиночном режиме или в режиме кластера горячего резервирования (см. «Система защиты от сбоев» на стр. 169). При работе в режиме

кластера доступны дополнительные команды для мониторинга и управления кластером. Описание всех команд, доступных в режиме кластера горячего резервирования, содержится в документе «ПАК ViPNet Coordinator HW. Система защиты от сбоев. Руководство администратора».

Команды группы `mftp`

- `mftp start` – запуск демона `mftpd`.
- `mftp stop` – остановка демона `mftpd`.
- `mftp info` – просмотр очереди конвертов `mftp`.
- `mftp view` – просмотр журнала `mftp`.

При выполнении этой команды производится постраничный показ лог-файла журнала `mftp`.

- `mftp config` – редактирование конфигурации демона `mftpd`.

Перед выполнением этой команды проверяется, остановлен ли демон `mftpd`, и если это не так, то выдается ошибка. Затем запускается текстовый редактор и в него загружается файл `mftp.conf` (см. «[Настройка конфигурации транспортного модуля](#)» на стр. 148). При сохранении файла происходит проверка его корректности и в случае ошибки предлагается отказаться от изменений или продолжить редактирование. Если проверка прошла успешно, файл применяется для работы демона `mftpd`, а информация об изменениях конфигурации сохраняется в журнале событий.

- `mftp show config` – просмотр файла конфигурации демона `mftpd`.

При выполнении этой команды не требуется остановка демона `mftpd`.



Примечание. Для завершения просмотра файла конфигурации нажмите клавишу «q».

Команды группы `admin`

- `admin passwd` – смена пароля пользователя.

После ввода этой команды запрашивается текущий пароль, причем можно ввести как пароль пользователя, так и пароль администратора. Если текущий пароль введен

правильно, то далее запрашивается новый пароль, а затем повторный ввод нового пароля. Если при повторном вводе новый пароль указан верно, то производится смена пароля пользователя.

При вводе паролей на экране ничего не отображается, введенные символы пароля отредактировать нельзя.

- `admin config save <имя>` – сохранение конфигурации под заданным именем (см. «Работа с конфигурациями ViPNet» на стр. 175).

При вводе имени конфигурации работают автозаполнение и подсказка, данные для подсказки берутся из списка существующих конфигураций.

- `admin config load <имя> [<версия>]` – загрузка конфигурации с заданным именем (см. «Работа с конфигурациями ViPNet» на стр. 175).

При вводе имени конфигурации работают автозаполнение и подсказка, данные для подсказки берутся из списка существующих конфигураций.

- `admin config delete <имя> [<версия>]` – удаление конфигурации с заданным именем (см. «Работа с конфигурациями ViPNet» на стр. 175).

При вводе имени конфигурации работают автозаполнение и подсказка, данные для подсказки берутся из списка существующих конфигураций.

- `admin config list [<версия>]` – вывод списка сохраненных конфигураций (см. «Работа с конфигурациями ViPNet» на стр. 175).

- `admin kick <имя терминала>` – посылка сигнала о завершении работы командного интерпретатора на указанном терминале.

Если на указанном терминале командный интерпретатор не запущен, то появляется сообщение об ошибке.

Если в команде указан собственный терминал (терминал, на котором введена данная команда), то появляется сообщение о том, что вместо этой команды следует использовать команду выхода `exit`.

- `admin export keys binary-encrypted <tftp | usb>` – экспорт ключей, справочников и настроек служб ViPNet на другой компьютер (по TFTP) или на USB-флэш.

Перед выполнением этой команды требуется остановить все демоны. Если не все демоны остановлены, то появляется сообщение о необходимости остановки демонов вручную, команда не выполняется.

Если все демоны остановлены, то появляется предупреждение о том, что во время экспорта ПАК будет полностью остановлен, и запрашивается подтверждение на выполнение команды. При получении подтверждения выполняется экспорт ключей,

справочников и настроек служб ViPNet (см. «Экспорт и импорт ключевых баз, справочников и настроек» на стр. 180).

Перед выполнением экспорта проверяется, запущен ли на ПАК DHCP-сервер. Если DHCP-сервер запущен, то он автоматически останавливается, а после завершения процедуры экспорта автоматически запускается.

- `admin remove keys` – удаление ключей и справочников.

При выполнении этой команды появляется предупреждение о том, что после удаления придется разворачивать ключи ViPNet заново, и запрашивается подтверждение на выполнение команды.

При получении подтверждения у пользователя запрашивается пароль администратора. Если введен верный пароль, то автоматически останавливается DHCP-сервер, если он запущен на интерфейсе eth1, и выполняется удаление ключей и справочников, после чего интерпретатор сообщает о том, что ключи ViPNet удалены, и завершает работу.

- `admin export logs <tftp | usb>` – экспорт протоколов работы, хранящихся на ПАК, на другой компьютер (по TFTP) или на USB-флэш (см. «Экспорт и импорт ключевых баз, справочников и настроек» на стр. 180).

Перед экспортом по TFTP (при указании параметра `tftp`) требуется остановить все демоны. Если не все демоны остановлены, то появляется сообщение о необходимости останова демонов вручную.

При указании параметра `usb` запрашивается подтверждение на выполнение экспорта.

- `admin remove logs` – удаление протоколов работы, хранящихся на ПАК.

Перед выполнением этой команды запрашивается подтверждение на удаление.



Примечание. Команды `admin export logs` и `admin remove logs` доступны только на модификациях ПАК с жестким диском, поддерживающих локальное хранение протоколов работы.

- `admin upgrade software usb` – обновление ПО ViPNet вручную с USB-флэш (см. «Локальное обновление ПО» на стр. 220).

- `admin escape` – выход в системную командную оболочку.

При вводе этой команды автозаполнение не работает.

После ввода команды появляется предупреждение о том, что данная команда предназначена для использования опытными администраторами в целях отладки и в

случае некорректных действий пользователя в системной командной оболочке компания «ИнфоТеКС» не гарантирует нормальную работу ПАК.

Затем запрашивается подтверждение на выполнение команды. При получении подтверждения у пользователя запрашивается пароль администратора. Если введен верный пароль, то запускается системная оболочка ОС Linux с правами пользователя `vipnet`.

После выхода пользователя из системной командной оболочки интерпретатор продолжит свою работу с той точки, в которой он находился перед запуском системной оболочки.

Команды группы `machine`

- `machine show timezone` – просмотр текущей временной зоны (часового пояса).
- `machine show date` – просмотр текущей даты и времени.
- `machine show uptime` – просмотр времени работы ПАК после перезагрузки, а также средней нагрузки.
- `machine show loghost` – просмотр текущего хоста, на который направляются протоколы работы.
- `machine show logs` – просмотр протоколов работы, хранящихся на ПАК.

Эта команда доступна только при локальном ведении протоколов работы (см. «[Журналы устранения неполадок ПО ViPNet](#)» на стр. 213).

- `machine show snmp-trapsink` – просмотр текущего хоста, на который посылаются оповещения по протоколу SNMP (см. «[Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP](#)» на стр. 210).
- `machine set timezone` – установка текущей временной зоны (часового пояса).

Установка текущей временной зоны производится так же, как при первоначальном развертывании ключевых баз (см. «[Первоначальное развертывание ключей](#)» на стр. 42). По команде последовательно выводятся списки континентов, стран и временных зон (часовых поясов), и в каждом из списков предлагается выбрать нужный элемент. Списки пронумерованы, для выбора нужного элемента надо ввести его номер (см. «[Списки континентов, стран и временных зон](#)» на стр. 163). Если какой-либо список содержит только один элемент, он выбирается автоматически. После выбора временной зоны выводится информация о текущем времени в этой зоне и запрашивается подтверждение на ее установку. При получении подтверждения в системе устанавливается выбранная временная зона. При

отрицательном ответе выводится список континентов (происходит возврат к началу установки).

- `machine set date <дата>` – установка текущей даты и времени (см. «[Настройка системного времени](#)» на стр. 161). Параметр задается в формате команды `date`. Дату и время следует указывать в формате `MMDDhhmm [YYYY]`, где:
 - `MM` –месяц;
 - `DD` -день;
 - `hh` –час;
 - `mm` –минуты;
 - `YYYY` -год, значение является необязательным. Если значение не установлено, используется значение по умолчанию (текущий год).

Перед установкой даты и времени требуется остановить все демоны, а после установки запустить их снова с помощью соответствующих команд.

- `machine set loghost <адрес>` – установка хоста, на который должны направляться протоколы работы.

Параметр `<адрес>` должен быть правильным IP-адресом, либо значением `null`, либо значением `local`. При указании значения `null` протоколирование не ведется. При указании значения `local` протоколы работы сохраняются в файле на жестком диске ПАК (см. «[Журналы устранения неполадок ПО ViPNet](#)» на стр. 213). Значение `local` можно указывать только на модификациях ПАК с жестким диском, поддерживающих локальное хранение протоколов работы.

- `machine set snmp-trapsink <адрес>` – установка хоста, на который должны направляться оповещения по протоколу SNMP (`snmp traps`).

Параметр `<адрес>` должен быть правильным IP-адресом либо значением `null` (см. «[Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP](#)» на стр. 210). При указании значения `null` протоколирование не ведется.

- `machine self-test` – запуск процедуры регламентного тестирования.

Перед запуском процедуры запрашивается подтверждение на ее выполнение. При получении подтверждения останавливаются все демоны. В процессе регламентного тестирования производится проверка целостности модулей и файлов конфигурации, проверка файловых систем на первом и втором разделах, проверка контрольных сумм ядра и образа ПО и т.д. При успешной проверке на консоль выводятся имена проверенных файлов и шестнадцатеричные значения их контрольных сумм. Если все проверки прошли успешно, то снова запускаются все демоны. При обнаружении ошибок ПАК выключается (автоматически выполняется команда `machine halt`).

- `machine reboot` – перезагрузка компьютера.

- `machine halt` – выключение компьютера.

Команды группы `ups`

- `ups set monitoring on` – включение взаимодействия ПАК с UPS.



Внимание! После включения взаимодействия ПАК с UPS необходимо вручную запустить демоны пакета NUT с помощью команды `ups start`.

- `ups set monitoring off` – отключение взаимодействия ПАК с UPS.

Взаимодействие может быть отключено, только если демоны пакета NUT остановлены. Если демоны запущены, запрашивается подтверждение на их остановку. В случае подтверждения сначала выполняется команда `ups stop`, затем взаимодействие отключается. В противном случае команда не выполняется.

- `ups set mode <master | slave <IP-адрес мастера>>` – задание режима взаимодействия ПАК с UPS (master или slave).

- `ups start` – запуск демонов пакета NUT.

Если демоны уже запущены, появляется сообщение об этом, команда не выполняется.

- `ups stop` – остановка демонов пакета NUT.

Если демоны уже остановлены, появляется сообщение об этом, команда не выполняется.

- `ups show config` – просмотр текущих настроек взаимодействия ПАК с UPS.

Выводится следующая информация:

- включено ли взаимодействие ПАК с UPS; остальная информация выводится только в случае, если взаимодействие включено;
- режим взаимодействия ПАК с UPS (master или slave);
- используемый драйвер UPS (только в режиме master);
- IP-адрес мастера (только в режиме slave);
- запущены или остановлены демоны пакета NUT.

- `ups show status [extended]` – просмотр текущего состояния UPS.

Выводится следующая информация:

- производитель UPS;
- модель UPS;
- текущая нагрузка по мощности (в процентах от максимальной нагрузки);
- текущий режим питания (OL - питание от сети, OB - питание от батареи);
- текущий уровень заряда батареи (в процентах от максимального уровня);
- расчетное время работы от батареи при текущих нагрузке и уровне заряда (в секундах);
- максимальное время работы от батареи, по истечении которого UPS посылает сигнал о необходимости отключения компьютера (задается производителем UPS).

При указании ключевого слова `extended` выводятся значения всех параметров состояния UPS в формате утилиты `upsc`, входящей в состав пакета NUT. Эта возможность предназначена для квалифицированных системных администраторов, которые могут самостоятельно интерпретировать результат вывода утилиты `upsc`.



Примечание. Если невозможно получить информацию о текущем состоянии UPS (например, если остановлен демон `upsd` на ПАК, находящемся в режиме `master`), то появляется сообщение об этом.

Прочие команды

- `version` – просмотр текущих версий продукта и компонентов.

Выводится версия продукта ПАК ViPNet Coordinator HW, номер поколения (G1 для первого поколения, G2 для второго поколения и т.п.), версия ПО ViPNet Coordinator Linux в составе ПАК, а также версии компонентов: демона `iplir`, драйвера `iplir`, демона `mftpd`, демона `failoverd`, командного интерпретатора.

- `who` – запрос информации о терминалах, на которых запущен командный интерпретатор.

Для каждого терминала, на котором запущен интерпретатор, выводится следующая информация: имя терминала, адрес подключения, время неактивности, режим работы интерпретатора (режим пользователя или администратора). Под неактивностью понимается отсутствие нажатия каких-либо клавиш при работе в интерпретаторе.

- `debug on [<facility.level>]` – включение вывода на консоль интерпретатора сообщений из журнала устранения неполадок, соответствующих указанным `facility` и `level`.

Если параметры `facility` и `level` не заданы, то по умолчанию подразумевается `daemon.debug`.

Если указаны `facility` и `level`, для которых вывод сообщений был включен ранее, то появляется сообщение об этом.

Сообщения из журнала некоторого демона выводятся на консоль интерпретатора только в случае, если в конфигурации этого демона для данных `facility` и `level` включено протоколирование (значение параметра `debuglevel` не равно -1).

Эту команду можно выполнить многократно с разными `facility` и `level`, при этом на консоль интерпретатора будут выводиться сообщения для всех `facility` и `level`, которые были последовательно заданы.

- `debug off [<facility.level>]` – отключение вывода на консоль интерпретатора сообщений из журнала устранения неполадок, соответствующих указанным `facility` и `level`.

Если параметры `facility` и `level` не заданы, то отключается вывод всех сообщений.

Если указаны `facility` и `level`, для которых вывод сообщений на консоль интерпретатора не был включен, то появляется сообщение об ошибке.

- `enable` – вход в режим администратора.

После ввода этой команды запрашивается пароль администратора. При вводе пароля на экране ничего не отображается, введенные символы пароля отредактировать нельзя.

Если пароль администратора введен верно, то выполняется переход в режим администратора.

- `exit` – выход из текущего режима: в режиме администратора – выход в режим пользователя, в режиме пользователя – завершение работы интерпретатора.



6

Настройка конфигурации управляющего демона

Общие принципы настройки	88
Настройка параметров защищенной сети	90
Настройка правил обработки открытых IP-пакетов	112
Настройка параметров сетевых интерфейсов	131
Работа с политиками безопасности	134
Настройка режимов работы через межсетевой экран	136
Настройка работы с удаленным Координатором через фиксированный альтернативный канал	142
Настройка ПАК, выполняющего функции Сервера Открытого Интернета	145

Общие принципы настройки

Настройка управляющего демона производится путем редактирования файлов конфигурации `iplir.conf`, `iplir.conf-<интерфейс>` и `firewall.conf`. Для этого в командном интерпретаторе подаются следующие команды:

- `iplir config` (для редактирования файла `iplir.conf`);
- `iplir config <интерфейс>` (для редактирования файла `iplir.conf-<интерфейс>`);
- `iplir config firewall` (для редактирования файла `firewall.conf`).

Перед редактированием файлов необходимо остановить управляющий демон командой `iplir stop`, произвести необходимые изменения и затем снова запустить управляющий демон командой `iplir start`. Остановка управляющего демона необходима потому, что он сам производит обновление информации в этих файлах по мере необходимости. Например, когда управляющий демон получает информацию от других Координаторов об изменениях IP-адресов Клиентов, он записывает обновленную информацию в файл `iplir.conf`. Поэтому перед редактированием этого файла управляющий демон должен быть остановлен.

Файлы `iplir.conf` и `iplir.conf-<интерфейс>` состоят из секций, каждая из которых содержит один или несколько параметров. Каждая строка содержит либо имя секции, заключенное в квадратные скобки, либо имя одного параметра вместе со значением. В имени секции нельзя использовать пробелы, табуляции и так далее. Строка с именем секции считается началом секции. Секция заканчивается там, где начинается следующая секция, или в конце файла. Все имена секций, параметров, названия протоколов и т.п., кроме имен сетевых узлов, должны быть написаны строчными буквами.

Любая строка в файле, начинающаяся с символа «#», считается комментарием пользователя и не учитывается при интерпретации файла. При обновлении файла управляющим демоном комментарии автоматически переносятся в начало секции, к которой они относятся.

Любая строка в файле, начинающаяся с символа «;», считается служебным комментарием. Эти строки добавляются в некоторых случаях автоматически при старте управляющего демона или в процессе его работы и служат для информирования пользователя о некритических ошибках конфигурации, о значении фильтров по каким-либо сервисам и т.д. Пользователю не следует добавлять служебные комментарии самостоятельно, они будут потеряны после следующего старта управляющего демона.

Каждая секция содержит один или несколько параметров. Имя параметра стоит первым словом в строке, за ним следует знак «=», затем пробел, затем значение параметра. Если значение состоит из нескольких частей, то они разделяются запятой, после которой следует пробел.

Файл конфигурации `firewall.conf` состоит из секций, содержащих одно или несколько правил обработки открытых IP-пакетов. Синтаксис файла `firewall.conf` (см. «[Настройка правил обработки открытых IP-пакетов](#)» на стр. 112) отличается от синтаксиса файлов `iplir.conf` и `iplir.conf-<интерфейс>`. Настройку правил обработки открытых IP-пакетов можно производить не только путем редактирования файла `firewall.conf`, но также с помощью апплета SGA (кроме правил обработки туннелируемых пакетов). Описание работы с апплетом SGA содержится в документе «Апплет мониторинга и управления ViPNet-координатором. Руководство пользователя».

Настройка параметров защищенной сети

Параметры настройки защищенной сети содержатся в файле `iplir.conf`. Для редактирования этого файла используется команда `iplir config`. Файл `iplir.conf` может содержать секции, описанные ниже.

Секция `[id]`

Секция `[id]` используется для описания адресных настроек и фильтров доступа к какому-либо защищенному сетевому узлу. Каждому узлу, с которым может связываться данный узел, соответствует своя секция `[id]`. Одна из секций `[id]` соответствует собственным настройкам ПАК (собственная секция).

Секция `[id]` содержит следующие параметры:

- `id` – уникальный идентификатор сетевого узла. По этому параметру управляющий демон отличает одну секцию `[id]` от другой. Идентификатор узла является единым для всей корпоративной сети, поэтому менять этот параметр нельзя. В каждой секции `[id]` может быть только один параметр `id`.
- `name` – имя данного узла. Этот параметр задается администратором сети ViPNet. Смысловой нагрузки он не несет и предназначен только для удобства настройки. Данный параметр записывается в файл конфигурации автоматически при его сохранении, редактировать его вручную не следует. В каждой секции `[id]` может быть только один параметр `name`.
- `group` – имя группы, в которую входит данный узел. Имя группы задается в случае необходимости использования альтернативного канала для взаимодействия ПАК с данным узлом (см. «[Настройка работы с удаленным Координатором через фиксированный альтернативный канал](#)» на стр. 142).

Все узлы, у которых присутствует параметр `group` и совпадает имя группы, считаются входящими в соответствующую группу. Узлы, у которых отсутствует данный параметр, считаются не входящими ни в какую группу. Именем группы может быть произвольная строка, состоящая из символов латинского алфавита, цифр и знаков дефис, подчеркивание и точка. В имени группы учитывается регистр символов.

Параметр `group` является необязательным и может указываться в каждой секции `[id]` только один раз; указание нескольких таких параметров считается ошибкой. Кроме того, запрещено использование данного параметра в секции `[id]` для собственного узла, а также в секциях `[id]` для служебных фильтров. По умолчанию параметр `group` отсутствует в секции `[id]`.

- `ip` – IP-адрес данного узла. В случае присутствия в секции `[id]` параметра `secondaryvirtual` (см. ниже) со значением `on`, через запятую после реального адреса указывается соответствующий ему виртуальный адрес, причем первым идет реальный адрес, а за ним виртуальный. Например: `ip= 192.168.201.10, 10.1.0.5`. Назначение виртуальных адресов более подробно описано в разделе [Общие принципы назначения виртуальных адресов](#) (на стр. 109).

В каждой секции `[id]` может быть несколько параметров `ip` в тех случаях, когда описываемый узел имеет несколько сетевых интерфейсов или несколько IP-адресов на интерфейсе. Первым должен быть указан ближайший адрес, по которому есть доступ к данному узлу. При автоматическом обновлении адресов ближайший адрес помещается первым автоматически. При изменении порядка следования адресов сохраняется их привязка к виртуальным адресам (при наличии `secondaryvirtual=on`), т.е. виртуальный адрес перемещается вслед за своим реальным. Если указан только реальный адрес (запятой в этом случае нет) и присутствует `secondaryvirtual= on`, то считается, что этому адресу еще не сопоставлен виртуальный.



Внимание! Менять виртуальный адрес вручную не следует, он назначается автоматически.

- `accessip` – текущий IP-адрес доступа к данному узлу со стороны собственного узла, т.е. тот IP-адрес, который в текущий момент используется ПАК для связи с данным узлом. Может принимать значение одного из реальных IP-адресов данного узла или значение виртуального IP-адреса, в зависимости от физической топологии сети и режимов функционирования (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23) собственного узла и данного узла. Этот параметр носит информативный характер и служит для того, чтобы администратор в процессе работы мог узнать, по какому IP-адресу следует обращаться к данному узлу в текущий момент.



Внимание! Менять параметр `accessip` вручную не следует, он определяется автоматически.

- `firewallip` – для всех секций `[id]`, кроме собственной, данный параметр определяет внешний IP-адрес доступа к данному узлу в случае, если этот узел находится за каким-либо межсетевым экраном (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23). При работе с узлом, установленным за межсетевым экраном, все направленные к нему зашифрованные пакеты инкапсулируются в единый тип – UDP с адресом назначения, равным адресу, указанному в параметре `firewallip`, и портом назначения, равным порту, указанному в параметре `port` (см. ниже). Распознавание используемого типа межсетевого экрана осуществляется по совокупности дополнительных параметров (`proxyid`, `dynamic_timeout` и т.д.) в этой же секции (см. ниже). Использование данного параметра для собственной секции `[id]` описано в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136). Каждая секция `[id]` имеет только один параметр `firewallip`. Если параметр установлен в значение `0.0.0.0`, то считается, что данный узел не находится за межсетевым экраном.
- `port` – для всех секций `[id]`, кроме собственной, данный параметр определяет порт назначения, на который следует посылать пакеты для данного узла, если он работает в одном из режимов с использованием межсетевого экрана (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23). Каждая секция `[id]` имеет только один параметр `port`. Использование данного параметра для собственной секции `[id]` описано в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136).
- `channelfirewallip` – внешний IP-адрес доступа к данному узлу для альтернативного канала в случае включения работы через указанный альтернативный канал (см. «[Настройка работы с удаленным Координатором через фиксированный альтернативный канал](#)» на стр. 142). Значение этого параметра состоит из 2-х частей, разделенных запятой: имя канала и внешний IP-адрес доступа для данного канала. В качестве имени канала должен быть указан один из каналов, определенных в секции `[channels]` (см. «[Секция \[channels\]](#)» на стр. 105). Указание любых других значений считается ошибкой. Кроме того, ошибкой считается указание в одной секции `[id]` нескольких параметров `channelfirewallip` с одинаковым именем канала, а также использование данного параметра в собственной секции `[id]` и в секциях `[id]` для служебных фильтров. Данный параметр является необязательным и по умолчанию отсутствует в секциях `[id]`.
- `channelport` – порт доступа к данному узлу для альтернативного канала в случае включения работы через указанный альтернативный канал (см. «[Настройка работы с удаленным Координатором через фиксированный альтернативный канал](#)» на стр. 142). Значение этого параметра состоит из 2-х частей, разделенных запятой: имя канала и порт доступа для данного канала. В качестве имени канала должен быть указан один из каналов, определенных в секции `[channels]` (см. «[Секция \[channels\]](#)» на стр. 105). Указание любых других значений считается ошибкой. Допускается указание данного параметра только в случае, если в этой же секции `[id]` указан

параметр `channelfirewallip` для этого же канала. Кроме того, ошибкой считается указание в одной секции `[id]` нескольких параметров `channelport` с одинаковым именем канала, а также использование данного параметра в собственной секции `[id]` и в секциях `[id]` для служебных фильтров. Данный параметр является необязательным и по умолчанию отсутствует в секциях `[id]`.

- `proxyid` – тип используемого режима работы данного узла через межсетевой экран (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23). Для всех секций `[id]`, кроме собственной, данный параметр может принимать различные значения в зависимости от установленного режима. Использование данного параметра для собственной секции `[id]` описано ниже (см. «[Настройка режимов работы через межсетевой экран](#)» на стр. 136). Для удобства восприятия идентификаторы записываются в шестнадцатеричном формате как в параметре `id`, так и в параметре `proxyid`, при этом перед значением ставится префикс `0x`. Каждая секция `[id]` имеет только один параметр `proxyid`.

В большинстве случаев значения параметров `firewallip`, `port` и `proxyid` определяются автоматически по информации, полученной от других узлов. Если по каким-то причинам данные параметры неизвестны, их можно задать вручную. Если для узла (кроме собственного узла) какой-либо из параметров `firewallip`, `port` и `proxyid` имеет нулевое значение, то он не записывается в конфигурационный файл.

- `dynamic_timeout` – период опроса (в секундах) ViPNet-координатора, выбранного в качестве межсетевого экрана для данного узла, для обеспечения пропуска входящего трафика через межсетевой экран (см. «[Настройка режима „С динамической трансляцией адресов“](#)» на стр. 139). Данный параметр присутствует во всех секциях `[id]`, кроме собственной.



Внимание! Менять параметр `dynamic_timeout` вручную не следует, он определяется автоматически.

- `usefirewall` – для всех секций `[id]`, кроме собственной, этот параметр отвечает за использование настроек работы через межсетевой экран с данным узлом. Он может принимать значение `on` или `off`. Если он установлен в `off`, то параметры `firewallip`, `port` и `proxyid` для этой секции игнорируются, и работа с данным узлом будет возможна только по одному из его реальных IP-адресов. Для собственного узла данный параметр определяет использование внешнего межсетевого экрана, т.е. если параметр выставлен в значение `off`, то внешний межсетевой экран использоваться не будет. Значение `on` должно использоваться для всех остальных режимов работы. Использование данного параметра для настройки режима работы собственного узла через межсетевой экран описано в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136).

- `fixfirewall` – присутствует только в собственной секции `[id]`. Определяет режим фиксации настроек работы собственного узла через межсетевой экран. Может принимать значение `on` или `off`. По умолчанию значение параметра `off`. Использование данного параметра описано в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136).
- `virtualip` – базовый виртуальный адрес данного узла. Назначение данного параметра описано в разделе [Общие принципы назначения виртуальных адресов](#) (на стр. 109). Каждая секция `[id]` имеет только один параметр `virtualip`.



Внимание! Менять базовый виртуальный адрес узла вручную не следует, он назначается автоматически.

- `forcereal` – позволяет всегда обращаться к данному узлу по его реальному адресу, даже в тех случаях, когда он должен быть виден только по виртуальному (о правилах адресации узлов см. ниже). Этот параметр может принимать значение `on` или `off`, значение `off` равноценно отсутствию этого параметра в секции (при этом параметр удаляется из секции при следующем запуске управляющего демона). Использовать этот параметр нужно с осторожностью, так как у сетевых узлов, которые видны по виртуальным адресам, могут совпадать реальные адреса (если эти узлы находятся в частных сетях).



Внимание! Если у двух узлов с совпадающими реальными адресами (но разными виртуальными) установить параметр `forcereal` в значение `on`, это приведет к непредсказуемым результатам.

- `always_use_server` – признак работы узла в режиме использования межсетевого экрана с динамическим NAT с направлением трафика через выбранный Координатор (см. «[Настройка режима „С динамической трансляцией адресов“](#)» на стр. 139). Параметр принимает значение `on` и присутствует только в случае работы данного узла в указанном режиме.



Внимание! Параметр `always_use_server` является служебным и менять его вручную не следует.

- `secondaryvirtual` – механизм назначения виртуальных адресов для данного узла. Этот параметр может принимать значение `on` или `off`, значение `off` равноценно отсутствию этого параметра в секции (при этом параметр удаляется из секции при следующем запуске управляющего демона). При включении данного параметра каждому реальному адресу сетевого узла, указанному в параметре `ip`, назначается виртуальный адрес (см. «[Общие принципы назначения виртуальных адресов](#)» на стр. 109). Если данный параметр отсутствует или его значение равно `off`, то механизм назначения виртуальных адресов для каждого реального адреса данного узла выключается, работа с узлом осуществляется только с использованием базового виртуального адреса, определяемого параметром `virtualip`. В этом случае все виртуальные адреса, присутствующие в параметрах `ip`, удаляются из файла конфигурации. По умолчанию данный параметр отсутствует.
- `filterdefault` – действие над пакетами, направленными к данному узлу или от него, по умолчанию, если они не подпадают под более конкретные фильтры. Параметр может принимать значение `pass` (пропускать пакеты) или `drop` (не пропускать пакеты). Каждая секция `[id]` имеет только один параметр `filterdefault`.
- `blockforward` – включение/отключение блокировки транзитных пакетов, идущих от данного узла либо к нему. Этот параметр может принимать значение `on` или `off`, значение `off` равноценно отсутствию этого параметра в секции. По умолчанию параметр отсутствует для всех узлов. При включении данного параметра все транзитные пакеты для данного узла блокируются с кодом 70 (см. «[Журнал регистрации IP-пакетов](#)» на стр. 201). Параметр разрешается использовать только в секциях для чужих узлов. Его указание в собственной секции, а также в главном и широковещательном фильтрах защищенной сети является ошибкой, при этом управляющий демон не запускается.
- `filtertcp` – правила для пропуска или блокировки пакетов TCP. Значение этого параметра состоит из четырех или пяти частей, разделенных запятыми. Первая часть содержит номер локального порта TCP-соединения, вторая – номер удаленного порта TCP-соединения. В обоих случаях можно указывать вместо одного номера порта диапазон, в этом случае начало и конец диапазона разделяются дефисом. Обратите внимание на то, что, в отличие от большинства распространенных сетевых экранов, в ViPNet эти номера портов не зависят от направления пакета, они не описывают номера портов отправителя и получателя, а всегда описывают локальный и удаленный порт, независимо от того, в каком направлении идет пакет. Третья часть описывает, что нужно делать с пакетом, и может принимать значение `pass` (пропускать) или `drop` (блокировать). Четвертая часть описывает направление TCP-соединения и может принимать значения `send`, `recv` и `any`. Эта часть также описывает не направление пакета, а направление TCP-соединения, т. е. какой узел являлся при установлении TCP-соединения клиентом, а какой – сервером. Если она принимает значение `send`, то из TCP-пакетов, имеющих номера локального и

удаленного портов, соответствующие указанным, под правило подпадают пакеты, относящиеся к соединениям, в которых данный узел являлся клиентом, при этом направление самих пакетов не имеет значения. Если она принимает значение `recv`, то под правило подпадают пакеты, относящиеся к соединениям, в которых данный узел являлся сервером. Наконец, при значении `any` под правило подпадают любые пакеты с соответствующими номерами портов. Пятая часть необязательна и может принимать значение `disable`, которое указывает на временное отключение данного фильтра, при этом он ведет себя так, как будто его не существует.

Например, правило

```
filtertcp= 22, 1024-65535, pass, recv
```

указывает, что должны пропускаться TCP-пакеты, относящиеся к тем соединениям, в которых удаленный узел, использующий номер порта от 1024 до 65535, установил соединение с портом 22 данного узла. При этом пакеты пропускаются в обоих направлениях. С другой стороны, это правило не разрешает пропускание пакетов, где данный узел устанавливает соединение с портом 22 удаленного узла.

- `filterudp` – правила для пропускания или блокировки пакетов UDP. Синтаксис этого параметра аналогичен синтаксису параметра `filtertcp` с одним отличием: поскольку протокол UDP не подразумевает установку соединений, то четвертая часть правила, описывающая направление, относится непосредственно к направлению, в котором идет пакет:
 - `send` для пакета, посланного с данного узла;
 - `recv` для пакета, посланного на данный узел;
 - `any` для обоих направлений.
- `filtericmp` – правила для пропускания или блокировки пакетов ICMP. Синтаксис этого параметра похож на синтаксис параметра `filterudp`, однако вместо номеров портов указывается тип и подтип сообщений ICMP: первая часть задает тип, а вторая – подтип. При их задании также можно указывать диапазоны. Остальные части правила имеют то же значение, что и для параметра `filterudp`.

Каждая секция `[id]` может содержать сколько угодно параметров `filtertcp`, `filterudp` и `filtericmp`. При приходе какого-либо пакета от данного узла или посылке пакета на него фильтры просматриваются в том порядке, в котором они указаны. Если пакет соответствует какому-либо правилу, то выполняется заданное этим правилом действие, и просмотр правил на этом прекращается. Если пакет не соответствует ни одному правилу, то выполняется действие, заданное параметром `filterdefault`.

- `filterip` – правила обработки пакетов, соответствующих протоколам IP, отличным от TCP, UDP и ICMP (например, IGMP и т.п.). Значение этого параметра состоит из двух частей, разделенных запятыми. В первой части указывается номер протокола

IP. Вторая часть описывает, что нужно делать с пакетом, и может принимать значение `pass` (пропускать) или `drop` (блокировать). Указание в этом правиле номеров протоколов ICMP, TCP и UDP (1, 6 и 17) считается ошибкой. Каждая секция `[id]` может содержать любое количество параметров `filterip`.

- `filterservice` – правила обработки пакетов, соответствующих какому-либо сервису. Сервис представляет собой именованную совокупность правил `filtertcp`, `filterudp`, `filtericmp` (см. ниже). Значение этого параметра состоит из двух или трех частей, разделенных запятыми. Первая часть – это имя сервиса, на соответствие которому будет проверяться пакет. Вторая часть описывает действие с пакетами – `pass` или `drop`. Третья часть необязательна, если она указывается, то может принимать только значение `inform`. Если третья часть указана, то после старта управляющего демона после данной строки `filterservice` вставляются служебные комментарии, которые описывают, какую именно совокупность правил представляет данный сервис.
- `tunnel` – незащищенные компьютеры, которые туннелируются данным узлом. Имеет смысл только для узла, являющегося Координатором. Значение этого параметра имеет вид: `<ip1>-<ip2> to <ip3>-<ip4>`, где `ip1` и `ip2` – начальный и конечный адреса туннелируемого диапазона, `ip3` и `ip4` – начало и конец отображения этого диапазона на данном узле. В большинстве случаев нужно задавать одинаковые диапазоны, то есть `ip3` такое же, как и `ip1`, и `ip4` такое же, как `ip2`. Однако иногда бывают случаи, когда диапазон `ip1-ip2` принадлежит к частной сети, и такие же адреса частной сети уже есть в локальной сети данного узла. В этом случае диапазон `ip1-ip2` отображается на другие, свободные адреса путем указания других значений `ip3` и `ip4`. Следует отметить, что значение `ip4` игнорируется и генерируется автоматически путем прибавления к `ip3` разницы между `ip2` и `ip1`.
Например:

```
tunnel= 192.168.201.5-192.168.201.10 to 192.168.201.5-192.168.201.10
```

задает, что данный Координатор туннелирует адреса с 192.168.201.5 по 192.168.201.10, которые отображаются на локальной машине без изменения;

```
tunnel= 192.168.201.5-192.168.201.10 to 192.168.202.5-192.168.202.10
```

задает, что данный Координатор туннелирует адреса с 192.168.201.5 по 192.168.201.10, которые отображаются на адреса с 192.168.202.5 по 192.168.202.10.

Параметры `tunnel` не рассылаются по сети. Это означает, что если какой-либо Координатор туннелирует группу незащищенных компьютеров, то другие узлы не получают информацию об этом автоматически. Необходимо вручную указать, что данный Координатор будет туннелировать данные компьютеры, на каждом узле, который будет работать с этими туннелируемыми компьютерами посредством ViPNet.

Подробные примеры настройки туннелей в типовых схемах приведены в Приложении.

Некоторые из секций [id] имеют специальное значение. Самая первая секция [id] описывает компьютер, на котором установлен ПАК ViPNet Coordinator HW. Путем изменения параметров этой секции можно изменять собственные настройки, которые затем рассылаются по сети.

Несколько замечаний, касающихся собственной секции [id]:

- Менять параметры `ip` не следует. Они автоматически заполняются при старте управляющего демона значениями, полученными от операционной системы.
- Изменяя параметры `usefirewall`, `firewallip`, `port`, `proxyid`, `fixfirewall` в совокупности с изменением параметров секции `[dynamic]` (см. ниже), можно установить различные режимы работы через межсетевой экран (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23). Описание настроек различных режимов функционирования собственного узла через межсетевой экран приведено в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136).
- Если собственный узел будет туннелировать какие-либо незащищенные компьютеры, то при указании параметра `tunnel` нужно всегда задавать одинаковые значения для реального диапазона адресов и диапазона отображения. В этом случае параметр `tunnel` должен иметь вид `tunnel= ip1-ip2 to ip1-ip2`. При несоблюдении этого правила диапазон отображения приводится в соответствие с реальным диапазоном автоматически.

С помощью параметров собственной секции [id] также настраиваются различные режимы работы ПАК ViPNet Coordinator HW (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23). Более подробно установка и настройка этих режимов описана ниже. При этом необходимо уделять внимание настройке маршрутизации. Следует соблюдать несколько правил:

- Если собственный узел туннелирует какие-либо компьютеры, то на всех туннелируемых компьютерах необходимо указать собственный узел в качестве `default gateway`.
- Если собственный узел будет работать хотя бы с одним узлом по виртуальному адресу, то у него должен быть настроен `default gateway`. Если `default gateway` не будет указан, то пакет может быть заблокирован системой на ранней стадии и вообще не дойти до драйвера.

- Если собственный узел используется как прокси-сервер, то на нем должна быть включена функция IP forwarding, то есть параметр `ipforwarding` в секции `[misc]` (см. «Секция `[misc]`» на стр. 103) должен быть установлен в значение `on`.

За первой секцией `[id]` следуют две секции, также имеющие специальное значение. Они отличаются от других значением параметра `id`, равным `0xffffffff` и `0xfffffefe`. Эти секции предназначены только для установки фильтров. Для этих двух секций имеют смысл только следующие параметры: `filtertcp`, `filterudp`, `filtericmp`, `filterip`, `filterdefault`. Секция с `id= 0xffffffff` отвечает за широкоэвещательные пакеты, посылаемые сетевыми узлами. Установкой соответствующих фильтров в этой секции можно запрещать или разрешать прохождение определенных типов широкоэвещательных пакетов. Секция с `id= 0xfffffefe` – это главный фильтр защищенной сети. Правила, указанные в нем, просматриваются до того, как начинается просмотр фильтров для конкретного узла. Если пакет подпадает под какое-либо правило, то он сразу пропускается или блокируется, и дальнейший анализ фильтров прекращается. Если же для главного фильтра пакет подпадает под правило `filterdefault` и он установлен в `pass`, то анализируются фильтры для конкретного узла. Установка `filterdefault` в `drop` для главного фильтра приведет к полному блокированию всех зашифрованных пакетов. Некоторый набор фильтров, необходимый для нормальной работы защищенной сети, устанавливается в главном фильтре автоматически и не может быть удален.

Для связи с узлами, описанными в параметрах секций `[id]`, могут использоваться реальные или виртуальные адреса. Независимо от режимов работы узлов, для связи с узлами, от которых собственный узел получает широкоэвещательные пакеты (`broadcast`), используются их реальные адреса. В иных ситуациях используемый тип адреса для связи с узлом определяется следующим образом:

- Если ПАК работает в режиме **Без использования межсетевого экрана** (см. «[Настройка режима „Без использования межсетевого экрана“](#)» на стр. 136), т.е. не стоит за внешним межсетевым экраном, то:
 - Для связи с Клиентами, работающими в режиме **Без использования межсетевого экрана**, т.е. не стоящими за какими-либо внешними межсетевыми экранами, используются реальные адреса.
 - Для связи с узлами, работающими в режиме **Координатор** и использующими ПАК как ViPNet-прокси, используются реальные адреса.
 - Для связи со всеми другими сетевыми узлами используются виртуальные адреса.
- Если ПАК работает в режиме **Координатор** (см. «[Настройка режима „Координатор“](#)» на стр. 136), т.е. стоит за каким-либо ViPNet-прокси, то:

- Для связи с Клиентами, работающими в режиме **Без использования межсетевого экрана**, т.е. не стоящими за какими-либо внешними межсетевыми экранами, используются реальные адреса.
- Для связи с Координаторами, кроме используемого ViPNet-прокси, всегда используются виртуальные адреса.
- Для связи с сетевыми узлами, стоящими за тем же интерфейсом того же самого ViPNet-прокси, что и ПАК (их `firewallip` равен `firewallip` ПАК), а также для связи с самим ViPNet-прокси, используются реальные адреса.
- Для связи с сетевыми узлами, стоящими за тем же самым ViPNet-прокси, что и ПАК, но за другим интерфейсом, а также работающими в режиме, отличном от режима **Без использования межсетевого экрана**, т.е. стоящими за какими-либо внешними межсетевыми экранами, используются виртуальные адреса.
- Если ПАК работает в режимах **Со статической трансляцией адресов** (см. «[Настройка режима „Со статической трансляцией адресов“](#)» на стр. 137) или **С динамической трансляцией адресов** (см. «[Настройка режима „С динамической трансляцией адресов“](#)» на стр. 139), т.е. стоит за каким-либо внешним межсетевым экраном, то:
 - Для связи с Клиентами, работающими в режиме **Без использования межсетевого экрана**, т.е. не стоящими за какими-либо внешними межсетевыми экранами, используются реальные адреса.
 - Для связи с сетевыми узлами, стоящими за тем же самым внешним межсетевым экраном, что и ПАК (их `firewallip` равен `firewallip` ПАК), используются реальные адреса.
 - Для связи со всеми другими сетевыми узлами используются виртуальные адреса.

Во всех описанных выше случаях, когда для доступа к сетевому узлу должен использоваться виртуальный адрес, при установке для этого узла параметра `forcereal` (см. выше) для доступа к нему всегда используется реальный адрес.

Описанные выше правила видимости сетевых узлов запоминать, как правило, не нужно. Текущий адрес доступа к сетевому узлу всегда отображается в параметре `accessip` соответствующей секции `[id]`, и любые обращения по сети к этому узлу должны производиться по этому адресу.

Секция `[adapter]`

Секции `[adapter]` описывают сетевые адаптеры, установленные на компьютере. Каждому адаптеру должна соответствовать своя секция `[adapter]`. Все пакеты на адаптерах, не описанных секциями `[adapter]`, блокируются. Если в файле `iplir.conf`

нет ни одной секции [adapter], то управляющий демон при старте получает от системы список адаптеров и автоматически создает секции [adapter]. В качестве параметров данной секции достаточно указать только параметры name и type (см. ниже). Параметр ip указывать не обязательно, так как его значение будет получено от системы при запуске. В процессе работы управляющий демон производит периодический опрос параметров известных ему адаптеров с интервалом времени, задаваемым параметром ifcheck_timeout секции [misc] (см. «Секция [misc]» на стр. 103). Если обнаруживается, что адаптер деактивирован в системе, то он деактивируется и в драйвере сетевой защиты ViPNet. Если адаптер активируется или изменяется его адрес, то управляющий демон загружает эти изменения в драйвер. Информация обо всех описанных событиях выводится в системный журнал.

В секции [adapter] указываются следующие параметры:

- name – системное имя адаптера, например, eth0, ppp0 и т.п.

Если в системе заданы несколько IP-адресов на одном адаптере и присутствуют одно или несколько псевдоустройств (eth0:0, eth0:1 и т.д.), то для управляющего демона и драйвера все они будут представлять одно физическое устройство с базовым именем (eth0).

- ip – IP-адрес адаптера.

Этот параметр заполняется автоматически и присутствует только для информационных целей. Для каждого адаптера может быть несколько параметров ip (например, если на адаптере используются дополнительные адреса).

- type – тип адаптера для ViPNet.

Этот параметр можно устанавливать в одно из значений: internal (внутренний) или external (внешний). Параметр заполняется, исходя из следующей логики:

- Если ПАК не стоит за внешним межсетевым экраном или за ViPNet-прокси, т.е. работает в режиме **Без использования межсетевого экрана** (см. «[Настройка режима „Без использования межсетевого экрана“](#)» на стр. 136), то типы всех адаптеров должны быть установлены в internal.
- Если ПАК стоит за внешним межсетевым экраном или за ViPNet-прокси, т.е. работает в режиме, отличном от режима **Без использования межсетевого экрана**, то тип адаптера, посредством которого ПАК будет связываться с узлом, выполняющим функции межсетевого экрана, устанавливается в external, типы остальных адаптеров – в internal.

Каждому адаптеру, описанному секцией [adapter], соответствует дополнительный файл конфигурации, который называется iplir.conf-<name>, где <name> – системное имя адаптера, указанное в параметре name его секции [adapter]. Настройка этого файла

конфигурации описана ниже (см. [«Настройка параметров сетевых интерфейсов»](#) на стр. 131).

Секция [dynamic]

Секция [dynamic] содержит параметры, необходимые для настройки режима **С динамической трансляцией адресов** (см. [«Настройка режима „С динамической трансляцией адресов“»](#) на стр. 139):

- `dynamic_proxy` – включение или выключение режима **С динамической трансляцией адресов** на ПАК. Этот параметр может принимать значение `on` или `off`, что соответствует включению или выключению данного режима. По умолчанию значение параметра `off`. Описание выбора различных режимов для собственного узла приведено в разделе [Настройка режимов работы через межсетевой экран](#) (на стр. 136).
- `firewallip` – внешний IP-адрес доступа к ПАК, работающему в режиме **С динамической трансляцией адресов**, со стороны других сетевых узлов.



Внимание! Параметр `firewallip` определяется автоматически, редактировать его вручную не следует.

- `port` – порт назначения, на который следует посылать пакеты для собственного узла, работающего в режиме **С динамической трансляцией адресов**.



Внимание! Параметр `port` определяется автоматически, редактировать его вручную не следует.

- `forward_id` – идентификатор Координатора, находящегося во внешней сети и используемого для организации входящих соединений собственным узлом, работающим в режиме **С динамической трансляцией адресов**. Идентификатор записывается в шестнадцатеричном формате, при этом перед значением ставится префикс `0x`. Описание установки данного параметра приведено в разделе [Настройка режима «С динамической трансляцией адресов»](#) (на стр. 139).
- `always_use_server` – включение или выключение режима, при котором любой трафик с внешними узлами направляется через Координатор, выбранный в параметре `forward_id` данной секции, т.е. все соединения с другими узлами будут

происходить только через внешний Координатор. Этот параметр может принимать значение `on` или `off`. По умолчанию значение параметра `off`. Описание установки данного параметра приведено в разделе [Настройка режима «С динамической трансляцией адресов»](#) (на стр. 139).

- `timeout` – период опроса (в секундах) Координатора для обеспечения пропуска входящего трафика через межсетевой экран. По умолчанию значение параметра 25 секунд. Описание установки данного параметра приведено в разделе [Настройка режима «С динамической трансляцией адресов»](#) (на стр. 139).

Секция [misc]

Секция [misc] содержит различные дополнительные параметры:

- `packettype` – формат шифрованных пакетов. В качестве формата пакетов могут быть выбраны 4.0 или 4.1. По умолчанию устанавливается формат 4.1.

Установка параметра `packettype` влияет только на формат пакетов, посылаемых данным сетевым узлом. Формат входящих пакетов определяется автоматически, и их расшифровка производится вне зависимости от установленного значения параметра `packettype`. Рекомендуется устанавливать формат 4.1, однако если необходимо связываться с узлами, на которых установлены старые версии ПО ViPNet, не поддерживающие формат 4.1, то необходимо установить формат пакетов 4.0.

- `ciphertype` – алгоритм шифрования для исходящих пакетов, адресованных сетевым узлам ViPNet. Параметр можно устанавливать только в значение `gost` (шифрование с помощью алгоритма ГОСТ).



Примечание. Установка параметра `ciphertype` влияет только на шифрование исходящего трафика.

- `timediff` – максимально допустимая разница во времени между сетевыми узлами. Из соображений безопасности ViPNet запрещает прохождение пакетов от сетевого узла, если его время отличается от времени собственного узла более, чем на число секунд, указанное в параметре `timediff`. По умолчанию его значение равно 7200, то есть два часа. При работе с узлами, находящимися в разных часовых поясах, это значение следует увеличить.

- `server_pollinterval`, `client_pollinterval` – интервалы времени опроса неактивных сетевых узлов. Сетевые узлы периодически обмениваются служебными пакетами, чтобы иметь информацию о том, работает какой-либо узел или нет. Клиенты обмениваются такой информацией только со своим Координатором – Сервером IP-адресов, а Координаторы – между собой. Параметр `server_pollinterval` отвечает за интервал опроса Координаторов со стороны данного узла. Параметр `client_pollinterval` отвечает за интервал опроса данного узла со стороны Клиентов, выбравших его в качестве Сервера IP-адресов, и сообщается им в каждом сеансе работы. Если от какого-либо узла, который должен обмениваться такими пакетами с данным узлом, не было получено никаких служебных пакетов в течение времени, указанного в соответствующем параметре `pollinterval`, то в адрес такого узла посылается специальный пакет, на который должен придти ответ. Если ответ не приходит, то узел считается выключенным. Значение параметров указывается в секундах и по умолчанию устанавливается в 900 секунд (15 минут) для Координаторов (`server_pollinterval`) и 300 секунд (5 минут) для Клиентов (`client_pollinterval`). Установка параметров в меньшие значения позволит более оперативно определять неработоспособность узла, но повысит объем служебного трафика, и наоборот.
- `iparponly` – включение или выключение блокировки пакетов, не принадлежащих IP-протоколу. ViPNet предназначен для анализа пакетов IP и по умолчанию пропускает все пакеты других протоколов, что соответствует установке параметра `iparponly` в значение `off`. Если установить этот параметр в значение `on`, то ViPNet будет блокировать пакеты всех протоколов, кроме IP, ARP и RARP. Пакеты протоколов ARP и RARP пропускаются всегда, поскольку их прохождение необходимо для успешного функционирования протокола IP.
- `ifcheck_timeout` – интервал опроса (в секундах) параметров адаптеров, известных управляющему демону. По умолчанию значение данного параметра 30 секунд.
- `warnoldautosave` – включение или выключение предупреждения о наличии старых автосохраненных конфигураций ViPNet (см. «Работа с конфигурациями ViPNet» на стр. 175). Может принимать значение `on` или `off`. Если значение параметра `on`, то при старте управляющего демона будут выдаваться предупреждения о наличии автоматически сохраненных конфигураций, дата сохранения которых меньше текущей даты более чем на один месяц.
- `ipforwarding` – управление включением IP-форвардинга в системе. Может принимать следующие значения:
 - `on` – принудительно включать IP-форвардинг при старте управляющего демона;
 - `off` – принудительно выключать IP-форвардинг при старте управляющего демона;
 - `system` – не изменять настройки форвардинга при старте управляющего демона.



Примечание. Рекомендуется выставлять значение `on`, так как при отключенном форвардинге не будет работать проксирование и туннелирование. Значения `off` и `system` рекомендуется использовать только при отладке.

- `mssdecrease` – число байт, на которое будет уменьшен параметр MSS (максимальный размер сегмента) протокола TCP для исключения фрагментации зашифрованных ViPNet-пакетов. Значение по умолчанию 0. В случае, если проверка соединения между узлами (`ping`) или туннелируемыми ресурсами проходит нормально, но TCP-соединения не устанавливаются, то, скорее всего, по пути следования пакетов на каких-то устройствах производится фрагментация пакетов и их блокировка. Для устранения таких проблем рекомендуется уменьшить значение MSS, например, на 20-40. Уменьшение параметра MSS достаточно произвести только с одной стороны. Данная настройка на Координаторе обеспечивает работоспособность как для узлов, взаимодействующих с Координатором, так и для туннелируемых устройств, стоящих за ним. Настройка на Координаторе не действует на узлы, стоящие за ним. Для таких узлов данный параметр следует изменять непосредственно на самих этих узлах или на других узлах, взаимодействующих с ними.



Внимание! Менять параметр `mssdecrease` без крайней необходимости не следует.

Секция [servers]

Секция `[servers]` содержит список Координаторов, известных ПАК. В этой секции присутствуют следующие параметры:

- `server` – описывает один из известных Координаторов. Значением каждого такого параметра является строка, где через запятую указаны идентификатор этого Координатора и его имя. Менять эти параметры не рекомендуется.

Секция [channels]

Секция `[channels]` определяет список альтернативных каналов доступа к ViPNet-координаторам со стороны данного узла, а также регистрацию групп Координаторов для работы через альтернативные каналы (см. «[Настройка работы с удаленным Координатором через фиксированный альтернативный канал](#)» на стр. 142). Секция

[channels] является необязательной, если она не задана, то считается, что ни один канал не определен. Если секция [channels] присутствует, но в ней нет ни одного параметра channel, то секция автоматически удаляется. Данная секция может содержать один или несколько параметров channel. Значение параметра channel должно иметь следующий формат:

```
<имя канала>, <список групп узлов, работающих через этот канал,  
разделенных запятыми>
```

Имя канала является его уникальным идентификатором. Имена групп узлов определяются в секциях [id] (см. «Секция [id]» на стр. 90). Пример задания параметров в этой секции:

```
[channels]  
channel= LocalNet, WorkDepartmentGroup, OtherGroup  
channel= ReservLocalNet  
channel= Internet, SalesDepartmentGroup
```

Именем канала может быть произвольная строка, состоящая из символов латинского алфавита, цифр и знаков дефис, подчеркивание и точка. В имени канала учитывается регистр символов. Указание списка имен групп после имени канала является необязательным.

Ошибками для конфигурации секции [channels] являются следующие:

- Для какого-либо параметра channel указана группа, которая не задана ни в одной секции [id].
- Для двух или более параметров channel заданы одинаковые имена групп, т.е. группа узлов не может одновременно быть зарегистрирована для нескольких каналов.
- Для двух или более параметров channel заданы одинаковые имена каналов.

Логика выбора работы с удаленным Координатором через заданный канал подробно описана ниже (см. «[Настройка работы с удаленным Координатором через фиксированный альтернативный канал](#)» на стр. 142).

Секция [service]

Секция [service] определяет один из сервисов. Сервис – это именованная совокупность фильтров, которые затем применяются для каких-либо сетевых узлов как единое целое, путем указания параметра filterservice (см. «Секция [id]» на стр. 90). В файле конфигурации может быть сколько угодно секций [service]. Рекомендуется, чтобы они предшествовали любым другим секциям; по крайней мере, они должны быть определены

раньше, чем будут использоваться в параметрах `filterservice`. После старта управляющего демона все определения сервисов автоматически помещаются в начало файла.

Секция `[service]` может содержать следующие параметры:

- `name` – имя сервиса. Имя может включать латинские буквы, цифры, а также знаки тире и подчеркивания. Этот параметр обязательно должен присутствовать, при этом имя сервиса должно быть уникальным в пределах данного файла конфигурации.
- `parent` – имя сервиса, наследником которого является данный сервис. Данный параметр необязательный. Если он указан, то его значением должно быть либо имя стандартного сервиса (см. ниже), либо имя сервиса, определенного выше в том же файле конфигурации. Сервис-потомок имеет все фильтры, определенные для его родителя, дополнительно к которым можно определять добавочные фильтры. Удалять какие-либо фильтры родителя в сервисе-потомке нельзя. Возможно указание нескольких родителей, в этом случае потомок наследует все их фильтры. Сервис-потомок можно использовать как родительский для других сервисов, при этом потомок получает всю цепочку фильтров, унаследованную его родителями. Глубина вложенности никак не ограничивается.
- `filtertcp`, `filterudp`, `filtericmp` – совокупность фильтров, которые будет содержать данный сервис. Синтаксис этих параметров такой же, как и соответствующих параметров в секции `[id]` (см. «Секция `[id]`» на стр. 90), с тем отличием, что в секции `[service]` у этих параметров не указывается третья часть, описывающая действие с пакетами – `pass` или `drop`. Эта часть просто оставляется пустой, и при указании данного сервиса в `filterservice` принимает то значение, которое указано в `filterservice`.

Пример описания сервиса:

```
[service]
name= http
filtertcp= 1024-65535, 80, , send
filtertcp= 80, 1024-65535, , recv
```

Секция [virtualip]

Секция [virtualip] содержит установки виртуальных адресов (см. «[Общие принципы назначения виртуальных адресов](#)» на стр. 109). В ней присутствуют следующие параметры:

- `startvirtualip` – стартовый адрес для генерации базовых виртуальных адресов. При смене пользователем параметра `startvirtualip` назначение всех базовых виртуальных адресов производится заново, как при начальном формировании файлов конфигурации. Кроме того, производится назначение виртуальных адресов в параметрах `ip` для узлов с `secondaryvirtual= on`.
- `maxvirtualip` – максимальный адрес для генерации базовых виртуальных адресов. Данный параметр используется для ограничения диапазона назначаемых базовых виртуальных адресов. По умолчанию параметр `maxvirtualip` соответствует максимально возможному адресу, то есть адресу, у которого два старших октета совпадают с этими же октетами стартового адреса `startvirtualip`, а два младших октета равны 254. Значение по умолчанию можно изменить в сторону уменьшения, при этом необходимо следить за тем, чтобы значение параметра `maxvirtualip` было больше, чем значение параметра `endvirtualip`.



Примечание. Параметр `maxvirtualip` поддерживается, начиная с версии 2.4. При обновлении более ранней версии на версию 2.4 или более позднюю версию в секцию [virtualip] автоматически добавляется параметр `maxvirtualip` со значением по умолчанию.

-
- `endvirtualip` – служебный параметр, в котором хранится следующий за последним назначенным базовый виртуальный адрес. Данный параметр используется в качестве точки отсчета при поиске и назначении базовых виртуальных адресов для новых узлов. При назначении базовых виртуальных адресов сначала производится поиск первого свободного адреса в диапазоне от `endvirtualip` по `maxvirtualip`. Если в этом диапазоне свободных адресов нет, то производится поиск в диапазоне от `startvirtualip` до `endvirtualip`.



Внимание! Менять параметр `endvirtualip` без крайней необходимости не следует, особенно в сторону увеличения.

-
- `startvirtualiphash` – служебный параметр.



Внимание! Менять параметр `startvirtualiphash` без крайней необходимости не следует, за исключением тонкой настройки с целью переназначения виртуальных адресов узлов (см. «[Ручное переназначение виртуальных адресов узлов](#)» на стр. 111).

Секция [debug]

Секция [debug] определяет параметры ведения журнала устранения неполадок управляющего демона (см. «[Журналы устранения неполадок ПО ViPNet](#)» на стр. 213). Она содержит следующие параметры:

- `debuglevel` – уровень протоколирования, число от -1 до 5. Для модификаций ПАК с жестким диском значение по умолчанию 3. Для ПАК ViPNet Coordinator HW100 базовой конфигурации (без жесткого диска) значение по умолчанию -1. Значение параметра -1 отключает ведение журнала.
- `debuglogfile` – место хранения журнала, заданное в виде `syslog:<facility.level>`. По умолчанию значение параметра устанавливается в `syslog:daemon.debug`.

Общие принципы назначения виртуальных адресов

Виртуальные адреса используются для сетевых узлов, находящихся за внешним межсетевым экраном. Необходимость использования виртуальных адресов обусловлена тем, что узлы, стоящие за разными межсетевыми экранами, могут иметь одинаковые адреса в своих частных сетях, и при использовании их реальных адресов при обращении к ним возникала бы неоднозначность. Для узлов, не находящихся за внешним межсетевым экраном, виртуальные адреса не используются, но все равно за каждым узлом закреплен свой виртуальный адрес. Исключения составляют Координаторы, работающие в режиме **Без использования межсетевого экрана** (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23).

Четыре октета, составляющие IP-адрес, используются при назначении виртуальных адресов следующим образом:

- Старший октет всех виртуальных адресов имеет одинаковое значение, соответствующее старшему октету начального виртуального адреса `startvirtualip`, заданного администратором.
- Два младших октета характеризуют сетевой узел. Они одинаковы для всех виртуальных адресов данного сетевого узла, и различны для виртуальных адресов, принадлежащих разным сетевым узлам.

- Второй по старшинству (слева) октет характеризует один из реальных адресов сетевого узла.

Другими словами, при проходе по сетевым узлам меняется сначала младший октет, затем следующий за ним (второй справа). При проходе по реальным адресам одного сетевого узла меняется второй слева октет. Старший октет в обоих случаях остается неизменным.



Примечание. Назначение виртуальных адресов для каждого реального IP-адреса узла осуществляется только при наличии у сетевого узла параметра `secondaryvirtual= on` (см. «Секция [id]» на стр. 90). В противном случае работа с узлом осуществляется только с использованием базового виртуального адреса.

Виртуальный адрес сетевого узла, в котором второй слева октет равен второму слева октету начального виртуального адреса `startvirtualip`, называется базовым виртуальным адресом `virtualip` сетевого узла. Базовый виртуальный адрес является точкой отсчета при назначении виртуальных адресов для каждого из реальных адресов узла.

Остальные виртуальные адреса сетевого узла, характеризующие каждый из реальных адресов узла, называются вторичными виртуальными адресами или для простоты – виртуальными адресами, и указываются в параметре `ip` через запятую после реального адреса. Один из виртуальных адресов узла может совпадать с базовым виртуальным адресом, что практически всегда будет в случае, если узел имеет один реальный адрес.

Как уже говорилось выше, текущий адрес доступа к сетевому узлу определяется автоматически и содержится в параметре `accessip`. Если в данный момент узел виден по виртуальному адресу, то его адресом доступа считается либо базовый виртуальный адрес, либо (в случае, когда `secondaryvirtual= on`) вторичный виртуальный адрес, соответствующий первому в списке реальному.

Вторичные виртуальные адреса могут использоваться, если нужно обратиться к конкретному реальному адресу данного сетевого узла, который виден по виртуальным адресам, а не к узлу вообще. Такая необходимость может возникнуть, например, если на данном сетевом узле работает приложение, которое ожидает сетевые запросы только на одном из адресов. В таких случаях нужно указывать параметр `secondaryvirtual= on`. В большинстве случаев для обращения к сетевому узлу достаточно одного виртуального адреса (базового), и указывать параметр `secondaryvirtual` следует лишь тогда, когда администратор узла четко понимает, для чего это нужно.

При обновлениях адресных справочников, а также изменениях в списке реальных адресов для какого-либо узла, сетевые узлы сохраняют свои виртуальные адреса, а вновь добавленные узлы и реальные IP-адреса получают новые свободные виртуальные адреса.

Ручное переназначение виртуальных адресов узлов

В стандартном режиме работы управляющий демон автоматически назначает виртуальные адреса (см. «[Общие принципы назначения виртуальных адресов](#)» на стр. 109) новым сетевым узлам, добавленным в связи. При этом виртуальные адреса для уже существующих узлов не изменяются. В случае удаления узла из связей соответствующий виртуальный адрес остается неиспользуемым, и образуется «дырка» в текущем диапазоне виртуальных адресов. В некоторых случаях данный алгоритм является неэффективным, например, если необходимо в силу определенных причин использовать ограниченный диапазон виртуальных адресов. В таком случае администратору необходим механизм, который позволит произвести повторное назначение виртуальных адресов узлам с заполнением «дырок», образовавшихся ранее при удалении связей с узлами.

Для повторного назначения виртуальных адресов узлов можно использовать параметр `startvirtualiphash` (см. «[Секция \[virtualip\]](#)» на стр. 108), который содержит хэш стартового виртуального адреса (параметр `startvirtualip`). С помощью данного параметра управляющий демон при старте определяет, был ли изменен стартовый виртуальный адрес, и в случае, если стартовый виртуальный адрес был изменен, производит переназначение виртуальных адресов для всех узлов. Чтобы повторно переназначить виртуальные адреса без изменения стартового виртуального адреса, необходимо остановить управляющий демон и удалить строку, содержащую параметр `startvirtualiphash`, а после этого снова запустить управляющий демон. В этом случае произойдет переназначение виртуальных адресов для всех сетевых узлов, начиная с адреса, указанного в параметре `startvirtualip`. При этом все образовавшиеся ранее «дырки» будут заполнены.

Настройка правил обработки открытых IP-пакетов

Правила обработки открытых (нешифрованных) IP-пакетов содержатся в файле `firewall.conf`.

Правила обработки включают в себя правила антиспуфинга, правила фильтрации IP-пакетов и правила трансляции адресов. Помимо правил, в файле `firewall.conf` содержатся служебные параметры межсетевого экрана.

Файл `firewall.conf` состоит из следующих секций:

- `[antispoof]` – правила антиспуфинга;
- `[local]` – правила фильтрации локальных IP-пакетов;
- `[broadcast]` – правила фильтрации широковещательных IP-пакетов;
- `[forward]` – правила фильтрации транзитных IP-пакетов;
- `[tunnel]` – правила фильтрации туннелируемых IP-пакетов;
- `[nat]` – правила трансляции адресов;
- `[settings]` – служебные параметры межсетевого экрана.

В качестве значений некоторых параметров в файле `firewall.conf` могут быть указаны следующие выражения:

- **Диапазон адресов** – два IP-адреса, разделенные дефисом. При задании диапазона второй адрес (конец диапазона) должен быть больше, чем первый адрес (начало диапазона). Диапазон адресов включает в себя все адреса, лежащие между началом и концом диапазона, а также начало и конец диапазона.

Например: `192.168.1.1-192.168.1.10`.

- **Диапазон идентификаторов** – два 32-битных идентификатора сетевых узлов, разделенные дефисом. При задании диапазона второй идентификатор (конец диапазона) должен быть больше, чем первый идентификатор (начало диапазона). Диапазон идентификаторов включает в себя все идентификаторы, лежащие между началом и концом диапазона, а также начало и конец диапазона.

Идентификаторы сетевых узлов записываются в шестнадцатеричном формате с префиксом 0x. Регистр букв A-F может быть любым, нули после префикса до первой значащей цифры могут быть опущены.

Например: 0x10e10000-0x10e100FF.



Примечание. Диапазон идентификаторов можно указывать только в правилах фильтрации туннелируемых пакетов.

- **Маска адресов** – сетевой адрес в формате CIDR (Classless Internet Domain Routing). Маска адресов состоит из адреса подсети в формате обычного IP-адреса и числа старших битов в маске подсети, которые равны 1. Адрес подсети и число битов разделяются символом «/» (прямой слеш).

Например: 192.168.1.0/24 (сеть с адресом 192.168.1.0 и маской подсети 255.255.255.0).

- **Маска идентификаторов** – идентификатор сетевого узла в формате, аналогичном CIDR. Маска идентификаторов состоит из 32-битного идентификатора ViPNet в шестнадцатеричном формате и числа старших битов в маске, которые равны 1. Идентификатор и число битов разделяются символом «/» (прямой слеш). Маска идентификаторов по смыслу аналогична маске адресов.

Например: 0x10e10000/16 (узлы с идентификаторами, у которых старшие 16 бит равны 4321 (0x10e1), т.е. все узлы, входящие в сеть ViPNet с номером 4321).



Примечание. Маску идентификаторов можно указывать только в правилах фильтрации туннелируемых пакетов.

- **Диапазон портов** – два номера портов (числа от 1 до 65535), разделенные дефисом. При задании диапазона второй номер (конец диапазона) должен быть больше, чем первый номер (начало диапазона). Диапазон портов включает в себя все порты, лежащие между началом и концом диапазона, а также начало и конец диапазона.

Например: 1024-65535.

Далее подробно описываются секции файла `firewall.conf`, а также синтаксис правил фильтрации пакетов и правил трансляции адресов.

Настройка правил антиспуфинга

Правила антиспуфинга позволяют задать для каждого интерфейса список IP-адресов, пакеты от которых допустимы на данном интерфейсе. При этом пакеты, которые не попадают в допустимый список, будут блокироваться. Кроме того, если на какой-либо интерфейс будут приходить пакеты с адресов, которые указаны как допустимые для другого интерфейса, то такие пакеты также будут блокироваться. Как видно из названия, основная задача антиспуфинга – это защита от так называемого «спуфинга», одного из видов сетевых атак, основанного на подделке IP-адреса. При спуфинге злоумышленник посылает какому-либо компьютеру пакет, в котором в качестве адреса отправителя указан не его собственный адрес, а какой-либо другой, который известен данному компьютеру. Например, таким образом можно послать пакет из Интернета на шлюз, задав в качестве адреса отправителя адрес частной внутренней сети, которая также подключена к данному шлюзу, при этом злоумышленник может получить доступ к какому-либо сервису, доступ к которому разрешен только из внутренней сети. Правила антиспуфинга позволяют исключить такую возможность.

Параметры антиспуфинга задаются в секции `[antispoof]` файла `firewall.conf`. Синтаксис этой секции такой же, как синтаксис файлов `iplir.conf` и `iplir.conf-<интерфейс>` (см. «Общие принципы настройки» на стр. 88).

Секция `[antispoof]` содержит следующие параметры:

- `antispoof` – включение или отключение механизма антиспуфинга. Этот параметр может принимать значение `yes` или `no`. По умолчанию значение параметра `no`.
- Параметры с именами, совпадающими с именами сетевых интерфейсов. Значением каждого параметра является список адресов, допустимых на данном интерфейсе. Список адресов может состоять из единичных адресов, диапазонов адресов, масок адресов и ключевых слов, указанных через запятую. Все адреса в списке должны принадлежать подсетям, подключенным к данному интерфейсу. В списке адресов можно указывать следующие ключевые слова:
 - `anypublic` – означает все адреса, допустимые в Интернете, т.е. все адреса, кроме выделенных для специальных целей: для локального сетевого интерфейса (127.0.0.0/8) и для частных сетей (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16);
 - `subnet` – означает всю подсеть, к которой принадлежит данный интерфейс, исходя из его адреса и маски подсети.

Если антиспуфинг включен, то при каждом старте управляющего демона или изменении параметров сети производится автоматическое формирование списка адресов, заданного ключевым словом `subnet`.

В секции антиспуфинга обязательно должны быть перечислены все сетевые интерфейсы, кроме локального (loopback). Локальный интерфейс не охватывается никакими правилами обработки пакетов, и на нем всегда пропускаются любые пакеты. Кроме того, следует отметить, что пакеты с адресов отправителя, лежащих в диапазоне 127.0.0.0/8, блокируются на всех интерфейсах, обрабатываемых ПО VIPNet, независимо от настроек антиспуфинга, поскольку таково требование стандартов.

При старте управляющего демона проверяется, включен ли антиспуфинг, и если он включен, проверяется наличие в секции [antispoof] всех интерфейсов, известных управляющему демону. Отсутствующие интерфейсы автоматически добавляются в секцию со значением subnet.

Пример секции антиспуфинга:

```
[antispoof]
antispoof= yes
eth0= anypublic
eth1= 192.168.1.0/24
```

В данном примере антиспуфинг включен и будет работать следующим образом:

- на интерфейс eth0 могут приходить пакеты со всех адресов, кроме частных (предполагается, что интерфейс подключен к Интернету);
- на интерфейс eth1 могут приходить пакеты с адресов от 192.168.1.1 до 192.168.1.255 (предполагается, что интерфейс подключен к локальной сети).

Если в данном примере на интерфейс eth0 из Интернета придет пакет с адресом отправителя из сети 192.168.1.0/24 либо 192.168.201.0/24, то он будет заблокирован. Таким образом, обеспечивается надежная защита от спуфинга.

Настройка правил фильтрации открытых IP-пакетов

Пакеты, прошедшие через механизм антиспуфинга, попадают на обработку правилами фильтрации открытых пакетов. Правила фильтрации открытых IP-пакетов задаются в секциях [local], [broadcast], [tunnel] и [forward] файла firewall.conf.

В секции [local] задаются правила фильтрации локальных пакетов – пакетов, у которых отправителем либо получателем является данный сетевой узел.

В секции [broadcast] задаются правила фильтрации широковещательных пакетов.

Наличие секций `[local]` и `[broadcast]` обязательно. При старте управляющего демона проверяется наличие данных секций в файле конфигурации. Если какой-либо из этих секций нет, то она автоматически добавляется в файл `firewall.conf` с правилами по умолчанию (см. «[Правила фильтрации открытых IP-пакетов по умолчанию](#)» на стр. 123).

В секции `[forward]` задаются правила фильтрации транзитных пакетов – пакетов, которые только проходят через данный сетевой узел на пути от отправителя к получателю. По умолчанию в файле `firewall.conf` присутствует пустая секция `[forward]`. Для возможности прохождения транзитных пакетов через узел необходимо, чтобы их прохождение было либо явно разрешено правилами в секции `[forward]`, либо соответствующие интерфейсы находились в режимах, позволяющих пропускать пакеты: интерфейс, на который приходят транзитные пакеты – в режиме 4, а интерфейс, с которого они уходят – в режиме 3 или 4. Если используются другие режимы, то нужно обязательно задать разрешающее правило в секции `[forward]`. Это относится также к случаю, когда используется трансляция адресов, более подробно настройка такой конфигурации описывается ниже.

В секции `[tunnel]` задаются правила фильтрации туннелируемых пакетов – пакетов, передаваемых между ресурсами, туннелируемыми данным узлом (Координатором), и защищенными узлами сети ViPNet. Наличие секции `[tunnel]` в файле конфигурации обязательно. При старте управляющего демона проверяется наличие данной секции. Если этой секции нет, то она автоматически добавляется в файл `firewall.conf` с правилом по умолчанию, которое разрешает трафик между всеми туннелируемыми ресурсами и всеми защищенными узлами, с которыми связан данный узел (см. «[Правила фильтрации открытых IP-пакетов по умолчанию](#)» на стр. 123).

В предыдущих версиях ПАК ViPNet Coordinator HW для пропуска туннелируемого трафика необходимо было задать в секции `[local]` разрешающие правила открытой сети. Для автоматического создания этих правил использовался параметр `autopasstunnels` в секции `[misc]` файла `iplir.conf`. В текущей версии этот параметр не поддерживается, а правила для туннелируемого трафика задаются в секции `[tunnel]`. При переходе на текущую версию параметр `autopasstunnels` автоматически удаляется из секции `[misc]` файла `iplir.conf`, а в файл `firewall.conf` добавляется секция `[tunnel]` с правилом по умолчанию. Уведомления о произведенных изменениях конфигурационных файлов выводятся в системный журнал.

Обработка IP-протоколов TCP/UDP/ICMP осуществляется на основании анализа различных параметров пакетов. С помощью правил фильтрации можно контролировать протокол, адрес и порт отправителя, адрес и порт получателя, направление установления соединения. Применение фильтрации пакетов по направлению установления соединения позволяет ограничить прохождение пакетов рамками установленных соединений: пропускать только запросы, инициализирующие соединения в заданном направлении, а также ответы на них, и запрещать запросы, инициализирующие соединения в обратном направлении.

Для обработки IP-протоколов, отличных от TCP/UDP/ICMP, создаются виртуальные соединения, основанные на IP-адресах и номере протокола. Таким образом, достаточно указывать разрешающее правило только для запроса, инициализирующего данное соединение, ответы будут приниматься автоматически (если они приходят с того же IP-адреса, на который отсылался запрос, и по тому же протоколу).

Каждая из перечисленных секций может содержать одно или несколько правил фильтрации. Синтаксис правил фильтрации одинаков для всех секций.

Каждое правило описывается параметром `rule`, его значение состоит из следующих компонентов:

```
rule= <управляющий компонент> <условие> <расписание> <действие>
```

или

```
rule= <управляющий компонент> <действие> <условие> <расписание>
```

Управляющий компонент должен указываться в самом начале правила, расписание должно указываться следом за условием.

Каждый из компонентов правила, в свою очередь, состоит из частей, которые называются лексемами. **Лексема** представляет собой служебное слово, после которого может указываться какой-либо параметр.

Компоненты правил, лексемы внутри компонентов, а также служебные слова и параметры внутри лексем отделяются друг от друга пробелами.

Управляющий компонент

Управляющий компонент описывает свойства правила, не относящиеся непосредственно к обработке пакетов, и всегда указывается в начале правила. Он может состоять из следующих лексем:

- `num <номер>` – указывает номер правила в секции (от 0 до 65535). Номера используются для обозначения приоритета правил – чем меньше номер, тем выше приоритет. При обработке пакета сначала проверяются условия тех правил, приоритет которых выше, и при совпадении условий выполняется указанное в правиле действие, после чего дальнейший просмотр правил прекращается.

Лексеме `num` можно не указывать, при этом ПО ViPNet попытается самостоятельно назначить правилу номер, исходя из номеров правил, которые находятся до и после данного правила. При этом не всегда получается тот результат, который ожидался, поэтому рекомендуется всегда явно указывать номер у каждого правила.

- `disable` – указывает на то, что данное правило временно отключено и не действует. Если указана лексема `num`, то лексема `disable` может указываться только после нее,

если лексемы `num` нет – то в начале правила. Отсутствие лексемы `disable` означает, что правило действует.

Условие

Условие описывает, какие параметры должен иметь пакет, чтобы он был обработан данным правилом. Условие может состоять из следующих лексем:

- `proto` <протокол> – указывает протокол транспортного уровня, к которому должен принадлежать пакет. Поддерживаются протоколы `tcp`, `udp` и `icmp`, также можно задавать цифровые номера любых протоколов. Если одно правило должно обрабатывать пакеты разных протоколов, то их надо указывать через запятую.

Вместо протокола можно указать ключевое слово `any`, что означает все протоколы.

В секции `[broadcast]` можно задавать только значения `udp` и `icmp`.

- `type` <тип> – указывает тип ICMP-сообщения. Эту лексему можно указывать только в условии для протокола ICMP (`proto icmp`), ее нельзя использовать для других протоколов и в случае указания всех протоколов (`proto any`). В лексеме `type` можно задать только один тип сообщения, который должен быть числом от 0 до 255.

Если в условии для протокола ICMP лексема `type` не указана, то считается, что под условие попадают любые типы ICMP-сообщений.



Примечание. Лексема `type` обязательно указывается в случае, если в условии для протокола ICMP указана лексема `code` (см. ниже).

- `code` <код> – указывает код ICMP-сообщения. Эту лексему можно указывать только в условии для протокола ICMP (`proto icmp`), ее нельзя использовать для других протоколов и в случае указания всех протоколов (`proto any`). В лексеме `code` можно задать только один код сообщения, который должен быть числом от 0 до 255.

Если в условии для протокола ICMP лексема `code` не указана, то считается, что под условие попадают любые коды ICMP-сообщений.



Примечание. В предыдущих версиях ПАК ViPNet Coordinator HW лексемы `type` и `code` не использовались. При переходе на текущую версию в условия правил, заданных для протокола ICMP, автоматически добавляется «`type 8 code 0`».

- `from <список адресов>` – описывает условия для адреса и порта отправителя пакета. Если указывается и адрес, и порт, то они разделяются двоеточием, например: `192.168.201.1:22`. Если порт не указывается, то двоеточие после адреса не ставится, в этом случае условие распространяется на все порты.



Примечание. В условии нельзя указывать номера портов для протокола `icmp` (`proto icmp`) и в случае указания всех протоколов (`proto any`).

Кроме единичного адреса, можно указать диапазон адресов или маску адресов, например: `192.168.1.1-192.168.1.10:22` или `192.168.201.0/24:22`. Можно также указывать диапазон портов, например, `192.168.201.0/24:1024-65535`. Можно перечислять несколько условий для адреса и порта через запятую, например: `192.168.1.1-192.168.1.10:22,172.16.1.0.24:25`.

Можно объединять адреса, диапазоны и маски адресов, а также порты и диапазоны портов в группы, перечисляя их через запятую и заключая группу в круглые скобки. Таким путем можно связать несколько диапазонов или масок адресов с одним диапазоном портов, не повторяя его несколько раз. Например, запись

`(192.168.201.0/24,172.16.1.0/24):1024-65535`

означает «пакеты со всех адресов в сетях `192.168.201.0/24` и `172.16.1.0/24`, имеющие порт отправителя от `1024` до `65535`». Возможны и более комплексные формы записи с одновременным группированием адресов и портов, а также с перечислением таких групп. Например, запись

`(192.168.201.0/24,172.16.1.0/24):(22,25,6660-6667),10.0.0.0/8:1024-65535`

означает «пакеты со всех адресов в сетях `192.168.201.0/24` и `172.16.1.0/24`, имеющие порт отправителя `22`, или `25`, или от `6660` до `6667`, а также пакеты с адресов из сети `10.0.0.0/8`, имеющие порт отправителя от `1024` до `65535`».

Вместо адресов и их диапазонов можно указывать следующие ключевые слова:

- `anyip` – означает все адреса (т.е. диапазон `0.0.0.0-255.255.255.255`);
- `broadcast` – означает адрес `255.255.255.255`.



Примечание. В предыдущих версиях ПАК ViPNet Coordinator HW вместо ключевого слова `anyip` в правилах использовалось слово `any`. При переходе на текущую версию слово `any` автоматически заменяется словом `anyip` там, где оно используется для указания всех IP-адресов.

- `to <список адресов>` – описывает условия для адреса и порта получателя пакета. Синтаксис этой лексемы такой же, как синтаксис лексемы `from`.

В секции `[broadcast]` в лексеме `to` можно указывать только следующие адреса:

- `broadcast` – означает адрес `255.255.255.255`;
 - `directed-broadcast` – означает широковещательные адреса всех подсетей, подключенных к интерфейсам компьютера. При загрузке правила в драйвер это значение заменяется на список соответствующих широковещательных адресов;
 - широковещательные адреса подсетей, подключенных к сетевым интерфейсам компьютера. Указание конкретного широковещательного адреса влияет на направленные широковещательные пакеты, посланные в соответствующей подсети.
- `in` или `out` – указывает направление установления соединения. При указании этих условий необходимо помнить о том, что задают не направление движения отдельного пакета, а именно направление установления соединения. ПО `VipNet` отслеживает, какие пакеты к каким установленным соединениям относятся, и пропускает их или отбрасывает в соответствии с правилами. Например, если в секции `[local]` задано условие

```
proto tcp from 192.168.1.1 to anyip out
```

то под такое условие попадают все локальные пакеты, относящиеся к тем соединениям, которые были инициированы с адреса `192.168.1.1`, т.е. пакеты, посланные с адреса `192.168.1.1` в адрес удаленных компьютеров, и ответные пакеты на адрес `192.168.1.1`. Однако если удаленный компьютер попытается установить соединение с `192.168.1.1`, то пакеты, относящиеся к такому соединению, не попадут под заданное условие.



Примечание. Для правил фильтрации транзитных и туннелируемых пакетов, задаваемых в секциях `[forward]` и `[tunnel]`, нельзя указывать направление установления соединения.

Для протокола TCP соединения отслеживаются всегда. Для протокола UDP, который не имеет понятия соединения, также производится попытка отслеживать виртуальное соединение, которое устанавливается в большинстве случаев между приложениями, использующими UDP. Например, если задано условие

```
proto udp from 192.168.1.1 to anyip:53 out
```

то после того, как компьютер с адресом `192.168.1.1` пошлет UDP-пакет на порт `53` удаленного компьютера, считается, что с ним установлено виртуальное соединение. Если затем в течение короткого времени придет ответ от удаленного компьютера на тот же порт, который использовался при отправке первого пакета, то такой ответ

также попадет под описанное выше условие, как принадлежащий к установленному виртуальному соединению. Виртуальные соединения считаются разорванными, если на них нет трафика в течение времени, определенного для данного протокола (см. «Служебные параметры межсетевого экрана» на стр. 127).

В тех случаях, когда приложения обмениваются пакетами UDP, используя разные номера портов для отсылки и приема пакетов, виртуальное соединение отследить невозможно. В таких случаях нужно рассматривать лексемы `in` и `out` как соответствующие направлению движения пакета.

Лексемы `proto`, `from` и `to` должны указываться в условии обязательно. Если какой-либо из этих параметров не важен, нужно указать `any` (в лексеме `proto`) или `anyip` (в лексемах `from` и `to`). Направление может не указываться, при этом считается, что под условие должны попадать соединения, устанавливаемые в обоих направлениях. Лексемы `type` и `code` в условии для протокола ICMP могут не указываться.

Примеры полных условий:

```
proto any from anyip to 192.168.201.1:22 in
proto tcp,udp from anyip:53 to 192.168.0.0/16,172.16.1.0/24
proto tcp from 10.0.0.1 to (192.168.0.0/16,172.16.1.0/24):(22,25) out
proto icmp type 8 code 0 from anyip to anyip
```

Особенности задания условия в правилах фильтрации туннелируемых пакетов

Для правил фильтрации туннелируемых пакетов, содержащихся в секции `[tunnel]`, условие задается с учетом следующих особенностей:

- В одной из лексем `from` и `to` должен быть указан список адресов туннелируемых ресурсов, в другой - список идентификаторов защищенных узлов, взаимодействующих с туннелируемыми ресурсами.
- Список идентификаторов составляется по тем же правилам, что и список адресов, например: `0x10e10000/16:(22,25)`. Для указания всех идентификаторов используется ключевое слово `anyid`.
- Вместо ключевых слов `anyid` и `anyip` можно использовать слово `any` (при этом можно указать порты). Если в одной из лексем `from` или `to` указано ключевое слово `any`, то в другой лексеме также должно быть указано слово `any`, иначе условие считается ошибочным. Такая запись заменяет собой 2 правила: первое правило, в котором в лексеме `from` указано слово `anyip`, а в лексеме `to` указано слово `anyid`, и второе правило, в котором в лексеме `from` указано слово `anyid`, а в лексеме `to` указано слово `anyip`.

Действие

Действие описывает, что нужно сделать с пакетом, параметры которого удовлетворяют условию. Действие задается одной из двух лексем:

- `pass` – указывает, что пакет должен быть пропущен.
- `drop` – указывает, что пакет должен быть отброшен (блокирован).

Расписание

Расписание позволяет задать временные интервалы, в течение которых действует правило. При отсутствии расписания правило действует постоянно. Расписание описывается одной лексемой `time`, параметр которой состоит из нескольких частей, разделенных запятыми:

```
time <режим расписания>,<тип расписания>,<интервал времени>
```

Режим расписания может принимать одно из следующих значений:

- `on` – означает, что правило действует в те интервалы времени, которые указаны в расписании, и не действует в остальное время;
- `off` – означает, что правило не действует в указанные интервалы времени и действует в остальное время (в противоположность значению `on`);
- `disable` – означает, что расписание отключается и правило действует всегда, как если бы расписания не было.

Тип расписания может принимать одно из следующих значений:

- `daily` – означает расписание на каждый день. Для этого типа расписания указывается один интервал времени, в течение которого правило будет включаться (если режим расписания `on`) или выключаться (если режим расписания `off`).
- `weekly` – означает расписание на неделю. Этот тип расписания позволяет задать для каждого дня недели свой интервал времени, в течение которого правило будет включаться (если режим расписания `on`) или выключаться (если режим расписания `off`). Такой тип расписания позволяет учитывать, например, другой режим работы для выходных дней.

Интервал времени в зависимости от типа расписания задается следующим образом:

- Если тип расписания `daily`, то задается один интервал в виде `hh:mm-НН:ММ`, где `hh:mm` – часы и минуты начала интервала, `НН:ММ` – часы и минуты конца

интервала. Время начала интервала включается в период действия расписания, а время конца – нет. Минуты могут принимать значения от 0 до 59, часы – от 0 до 24. Если число часов равно 24, то число минут может быть равно только 00, такое время означает 0 часов следующего дня.

- Если тип расписания `weekly`, то можно задать несколько интервалов времени. Для каждого дня недели указываются первые три буквы его английского названия (`mon`, `tue`, `wed`, `thu`, `fri`, `sat`, `sun`), затем знак равенства и временной интервал для данного дня недели в том же формате, что и для ежедневного расписания, например: `mon=9:00-18:00`. Расписания на разные дни недели разделяются запятыми, например: `mon=9:00-18:00,tue=10:00-18:00`. Кроме того, можно задать один и тот же интервал для нескольких дней недели, в этом случае дни недели перечисляются до знака равенства через двоеточие, например: `sat:sun=00:00-24:00`.

В расписании можно указывать не все дни недели. В этом случае в те дни, для которых расписание не задано, правило будет вести себя так же, как и в указанные дни за пределами заданных интервалов времени (т.е. выключаться, если режим расписания `on`, и включаться, если режим расписания `off`).

Примеры полных расписаний:

```
time off,daily,9:00-18:00
```

```
time on,weekly,mon:tue:wed:thu:fri=9:00-18:00,sat:sun=00:00-24:00
```

Правила фильтрации открытых IP-пакетов по умолчанию

Файл `firewall.conf` создается автоматически в процессе установки ПО на ПАК ViPNet Coordinator HW. Созданный файл содержит обязательные секции с заданными в них правилами по умолчанию. Некоторые из этих правил отключены, причем вместо лексемы `disable` для их отключения используется комментирование соответствующих строк. Для включения какого-либо правила достаточно удалить из строки знак комментария.

В процессе работы администратор может изменять набор правил, однако всегда можно вернуться к правилам по умолчанию в любой из обязательных секций. Для этого надо целиком удалить секцию из файла `firewall.conf`. При следующем запуске управляющего демона отсутствующая секция будет автоматически добавлена в файл `firewall.conf` с правилами по умолчанию.

В секции `[local]` по умолчанию присутствуют следующие правила:

```
rule= proto udp from anyip:67 to anyip:68 pass
rule= proto udp from anyip:68 to anyip:67 pass
# rule= proto udp from anyip:138 to anyip:138 pass
rule= proto udp from anyip to anyip:53 pass
```

```
rule= proto udp from anyip to anyip:123 pass
```

Первые два правила разрешают пропуск пакетов службы DHCP (порты 67 и 68), предназначенной для динамического выделения компьютерам IP-адресов. Эти правила включены (действуют).

Третье правило разрешает пропуск пакетов службы датаграмм NetBIOS (netbios-dgm, порт 138), предназначенной для передачи данных между компьютерами при использовании в локальной сети NetBIOS-имен. Это правило отключено (закомментировано).

Четвертое правило разрешает пропуск исходящих пакетов на 53-й порт DNS-серверов, пятое правило разрешает пропуск исходящих пакетов на 123-й порт NTP-серверов. Эти правила включены (действуют).

В секции [broadcast] по умолчанию присутствуют следующие правила:

```
rule= proto udp from anyip:67 to anyip:68 pass
rule= proto udp from anyip:68 to anyip:67 pass
# rule= proto udp from anyip:138 to anyip:138 pass
# rule= proto udp from anyip:137 to anyip:137 pass
```

Первые три правила те же, что и в секции [local]. Четвертое правило разрешает пропуск пакетов службы имен NetBIOS (netbios-ns, порт 137), предназначенной для регистрации и проверки NetBIOS-имен компьютеров в локальной сети. Это правило отключено (закомментировано).

В секции [tunnel] по умолчанию присутствует следующее правило:

```
rule= proto any from any to any pass
```

Это правило включено и разрешает трафик между всеми туннелируемыми ресурсами и всеми защищенными узлами, с которыми связан ПАК. Такая запись эквивалентна заданию 2-х следующих правил для туннелируемых ресурсов:

```
rule= proto any from anyid to anyip pass
rule= proto any from anyip to anyid pass
```

Настройка правил трансляции адресов

ПАК ViPNet Coordinator HW поддерживает трансляцию адресов, то есть изменение адреса отправителя или получателя пакета по определенным алгоритмам.

Поддерживаются два типа трансляции адресов:

- **Трансляция адреса отправителя**, называемая также маскардингом или динамической трансляцией. Такая трансляция адресов используется, если нужно

организовать выход в Интернет пользователей, имеющих частные адреса. В этом случае при проходе через ПАК пакетов от частных отправителей в них заменяется адрес отправителя на внешний (реальный) адрес ПАК. При приходе ответных пакетов в них подменяется адрес получателя обратно на частный адрес, и в таком виде пакет доставляется в частную сеть.

- **Трансляция адреса получателя**, называемая также форвардингом портов или статической трансляцией, используется, когда нужно обеспечить доступ из Интернета к компьютеру, находящемуся в частной сети. В этом случае пакеты, приходящие из Интернета на определенный порт внешнего адреса ПАК, перенаправляются на указанный адрес внутренней сети путем подмены в них адреса получателя, а у ответных пакетов от компьютера внутренней сети подменяется адрес отправителя.

Правила трансляции адресов задаются в секции [nat] файла `firewall.conf`.

Синтаксис правил трансляции адресов

Правила трансляции адресов имеют синтаксис, подобный синтаксису правил фильтрации (см. «[Настройка правил фильтрации открытых IP-пакетов](#)» на стр. 115). Каждое правило описывается параметром `rule`, его значение состоит из следующих компонентов:

```
rule= <управляющий компонент> <действие> <условие>
```

Управляющий компонент должен указываться в самом начале правила, остальные компоненты правила могут следовать в любом порядке.



Примечание. В отличие от правил фильтрации, в правилах трансляции адресов отсутствует расписание.

Управляющий компонент полностью идентичен управляющему компоненту, описанному для правил фильтрации (см. «[Настройка правил фильтрации открытых IP-пакетов](#)» на стр. 115).

Действие описывается лексемой `change` с параметром, указывающим, что именно нужно подменять и на что именно. В зависимости от того, какой тип трансляции адресов нужно осуществлять, действие может принимать следующий вид:

- Для трансляции адреса отправителя: `change src=<адрес>:dynamic`
где `<адрес>` – внешний адрес ПАК, на который будет заменяться адрес отправителя пакетов. Например: `change src=194.87.0.8:dynamic`.

- Для трансляции адреса получателя: `change dst=<адрес>:<порт>`
где `<адрес>` и `<порт>` – адрес и порт компьютера в локальной сети, на который будет производиться перенаправление пакетов. Например: `change dst=192.168.201.1:8080`.

Условие для правил трансляции адресов имеет такой же синтаксис, как и для правил фильтрации, но с некоторыми особенностями:

- Для трансляции адреса отправителя (динамическая трансляция адресов) в лексеме `proto` должно быть указано `any`, а в лексеме `to` должно быть указано `anyip`.
- Для трансляции адреса отправителя лексема `from` указывает набор адресов внутренней сети, которые будут подвергаться трансляции, при этом в лексеме `from` можно указывать только адреса, диапазоны адресов и маски, а также их списки. Указывать порты или диапазоны портов нельзя.
- Для трансляции адреса получателя (статическая трансляция адресов) в лексеме `from` должно быть указано `anyip`, а в лексеме `to` должны быть указаны внешний адрес и порт ПАК, на который будут приходить пакеты, подлежащие перенаправлению. В этом случае в лексеме `to` можно указывать только адрес или список адресов, а также порт или список портов. Указание диапазонов и масок адресов, а также диапазонов портов запрещено.

Примеры полных правил трансляции адресов:

- Для трансляции адреса отправителя:
`rule= num 10 change src=194.87.0.8:dynamic proto any from 192.168.201.0/24 to anyip`
- Для трансляции адреса получателя:
`rule= num 100 change dst=10.0.0.7:8080 proto tcp from anyip to 194.87.0.8:80`

Взаимодействие правил фильтрации и правил трансляции

Пакеты, подвергающиеся преобразованию адресов, проходят также обработку правилами фильтрации, указанными в секциях `[local]` и `[forward]`. При установлении соответствия между параметрами пакета, прошедшего через трансляцию адресов, и условием правила фильтрации применяется следующий принцип: адрес отправителя берется из пакета до трансляции, а адрес получателя – после трансляции. Именно эти адреса проверяются на соответствие условиям правил.

Например, пусть существует ПАК с внутренней сетью 10.0.1.0/24 и внешним адресом 194.87.0.8. Необходимо обеспечить доступ пользователей внутренней сети в Интернет. Для этого в секции [nat] должно быть задано следующее правило:

```
rule= num 10 change src=194.87.0.8:dynamic proto any from 10.0.1.0/24 to anyip
```

Необходимо также задать разрешающее правило в секции [forward]:

```
rule= num 100 pass proto any from 10.0.1.0/24 to anyip
```

Если при этом необходимо запретить внутренним пользователям устанавливать TCP-соединения с внешним адресом 194.226.82.50, то для этого в секцию [forward] нужно добавить следующее правило:

```
rule= num 90 drop proto tcp from 10.0.1.0/24 to 194.226.82.50
```

Как видно из примера, в лексеме *from* указан адрес локального отправителя пакета до трансляции адресов, а в лексеме *to* – адрес удаленного получателя после трансляции адресов (в данном случае он не изменяется в процессе трансляции).

Тот же принцип применяется при трансляции адреса получателя. Пусть необходимо перенаправлять все пакеты, приходящие на порт 80 внешнего адреса ПАК, в локальную сеть на адрес 10.0.1.1 и порт 8080. Для этого в секции [nat] должно быть задано следующее правило:

```
rule= num 10 change dst=10.0.1.1:8080 proto tcp from anyip to 194.87.0.8:80
```

Необходимо также задать следующее разрешающее правило в секции [forward]:

```
rule= num 100 pass proto tcp from anyip to 10.0.1.1:8080
```

Если при этом необходимо запретить входящие соединения на данный компьютер внутренней сети с внешнего адреса 194.226.82.50, то для этого в секцию [forward] нужно добавить следующее правило:

```
rule= num 90 drop proto tcp from 194.226.82.50 to 10.0.1.1:8080
```

Служебные параметры межсетевого экрана

Параметры межсетевого экрана, не являющиеся правилами обработки открытых IP-пакетов, задаются в секции [settings] файла `firewall.conf`. Синтаксис этой секции такой же, как синтаксис файлов `iplir.conf` и `iplir.conf-<интерфейс>` (см. «[Общие принципы настройки](#)» на стр. 88). При создании файла `firewall.conf` секция [settings] не содержит параметров, при этом используются значения по умолчанию, приведенные ниже.

Секция `[settings]` содержит следующие параметры:

- `max-connections` – максимальное количество одновременных соединений. Следует иметь в виду, что число обрабатываемых реальных физических соединений будет в 3 раза меньше. Значение по умолчанию 300000, это максимально допустимое значение параметра. Если необходимо ограничить число одновременных соединений, следует уменьшить значение параметра.
- `dynamic-ports` – диапазон портов, которые используются для динамической трансляции адресов. Значение по умолчанию 60000–65000.
- `listen-timeout` – время ожидания «вспомогательных» соединений (в секундах).
Некоторым протоколам во время работы требуется установка дополнительных соединений. Если установлен соответствующий модуль обработки прикладных протоколов, то автоматически создаются правила для вспомогательных соединений. Эти правила удаляются, если в течение времени, заданного в `listen-timeout`, не было установлено соединение. Значение по умолчанию 3 (секунды).
- `connection-ttl-tcp` – время (в секундах), по истечении которого после регистрации последнего пакета, относящегося к данному соединению TCP, оно разрывается по тайм-ауту. Значение по умолчанию 3600 (60 мин).
- `connection-ttl-udp` – время (в секундах), по истечении которого после регистрации последнего пакета, относящегося к данному соединению UDP, оно разрывается по тайм-ауту. Значение по умолчанию 300 (5 мин).
- `dynamic-timeouts` – включение или отключение режима динамических тайм-аутов соединений (`yes/no`). Значение по умолчанию `no`.

Режим динамических тайм-аутов используется для противодействия flood-атакам и работает следующим образом: когда количество соединений достигает определенного процента от максимума, тайм-ауты всех соединений уменьшаются на определенную величину, и эта величина тем больше, чем ближе число соединений к максимуму (но тайм-аут не уменьшается ниже определенного минимума). Когда количество соединений падает до определенного процента от максимального, тайм-ауты восстанавливаются до нормальной величины.

- `cleanup-interval` – периодичность удаления устаревших соединений (с истекшими тайм-аутами). Задается в секундах, значение по умолчанию 5 (секунд).

Большие значения приведут к менее аккуратному (по времени) удалению устаревших соединений, а слишком маленькие к лишней нагрузке на процессор.

Настройка правил фильтрации и трансляции с помощью апплета SGA

Настройку ПАК ViPNet Coordinator HW можно осуществлять не только путем редактирования файлов конфигурации, но также с помощью апплета SGA. Апплет SGA предназначен для мониторинга и управления ViPNet-координатором посредством веб-интерфейса и позволяет, в частности, настроить правила фильтрации открытых IP-пакетов и правила трансляции адресов. Подробное описание работы с апплетом SGA содержится в документе «Апплет мониторинга и управления ViPNet-координатором. Руководство пользователя».



Примечание. Правила фильтрации туннелируемых IP-пакетов нельзя просмотреть и изменить с помощью апплета SGA.

Настройка правил фильтрации и трансляции с помощью апплета SGA имеет некоторые особенности, обусловленные тем, что представление правил в интерфейсе апплета отличается от их записи в файле `firewall.conf`. В файле конфигурации можно указывать комплексные формы записи условий (списки протоколов, IP-адресов и т.д., группировка условий), тогда как в апплете SGA возможны только простые формы представления (единичные протоколы и адреса, диапазоны адресов и т.д.). Если файл `firewall.conf` содержит сложные правила (в которых указаны сложные условия), то при просмотре правил с помощью апплета SGA каждое сложное правило преобразуется и/или подвергается декомпозиции, т.е. разбивается на несколько простых правил. В случае, когда апплет SGA используется только для просмотра правил, записи в файле `firewall.conf` не изменяются. Если правила редактируются и затем сохраняются с помощью апплета SGA, то в файле `firewall.conf` изменяются записи, относящиеся к сложным правилам.



Примечание. При редактировании и сохранении правил фильтрации и трансляции с помощью апплета SGA сложные правила разбиваются на несколько простых правил, так что в файле `firewall.conf` вместо одного сложного правила будут записаны несколько более простых правил.

Декомпозиция правил фильтрации открытых IP-пакетов проводится в следующих случаях:

- в правиле фильтрации указан список протоколов;
- в правиле фильтрации указан список IP-адресов;

- в правиле фильтрации указан список портов отправителя и/или список портов получателя;
- в правиле фильтрации указана маска для IP-адресов.

Например, если в файле `firewall.conf` в секции `[local]` задано правило

```
rule= num 1 proto any from 10.0.2.0/24 to anyip out pass
```

то после сохранения правил фильтрации с помощью апплета SGA это правило будет преобразовано следующим образом:

```
rule= num 1 proto any from 10.0.2.0-10.0.2.255 to anyip out pass
```

В данном примере маска подсети, заданная в условии правила, преобразуется в диапазон адресов.

Декомпозиция правил трансляции адресов проводится в следующих случаях:

- в правиле трансляции адреса отправителя (динамическая трансляция адресов) в лексеме `from` указан список IP-адресов отправителя и/или маски для IP-адресов;
- в правиле трансляции адреса получателя (статическая трансляция адресов) в лексеме `to` указан список IP-адресов и/или портов получателя.

Например, если в файле `firewall.conf` в секции `[nat]` задано правило

```
rule= num 1 proto tcp from anyip to 80.251.135.75:(80,81) change
dst=10.0.2.1:8080
```

то после сохранения правил трансляции с помощью апплета SGA вместо этого правила в файле `firewall.conf` будут записаны 2 правила:

```
rule= num 1 proto tcp from anyip to 80.251.135.75:80 change
dst=10.0.2.1:8080
```

```
rule= num 1 proto tcp from anyip to 80.251.135.75:81 change
dst=10.0.2.1:8080
```

В данном примере одно правило, в условии которого указан список из 2-х портов, разбивается на 2 правила, в условиях которых указан только один порт.

Настройка параметров сетевых интерфейсов

Параметры настройки сетевых интерфейсов содержатся в файлах `iplir.conf-
<интерфейс>`. Для редактирования этих файлов используется команда `iplir config
<интерфейс>`. Файлы конфигурации для интерфейсов содержат секции, описанные ниже.

Секция [mode]

Секция [mode] используется для задания режима безопасности драйвера ViPNet на данном интерфейсе. Эта секция содержит следующий параметр:

- `mode` – номер режима безопасности на интерфейсе. Может принимать следующие значения:
 - 1 – блокировать IP-пакеты всех соединений (пропускать только зашифрованные пакеты);
 - 2 – блокировать все соединения, кроме разрешенных (пропускать зашифрованные пакеты, а также незашифрованные пакеты от адресатов, явно указанных в фильтрах открытой сети);
 - 3 – пропускать все исходящие соединения, кроме запрещенных (правило бумеранга: в дополнение к режиму 2 пропускать все исходящие пакеты, а также (в течение некоторого времени) входящие пакеты от узлов, соединение с которыми установлено исходящими пакетами);
 - 4 – пропускать все соединения;
 - 5 – пропускать IP-пакеты без обработки (отключить драйвер, не расшифровывать зашифрованные пакеты).

По умолчанию на всех интерфейсах ПАК установлен режим безопасности 2.

Подробное описание режимов безопасности приведено в разделе [Основные режимы безопасности ПО ViPNet](#) (на стр. 21).

Секция [db]

Секция [db] используется для задания параметров журнала трафика. Журнал ведется отдельно для каждого интерфейса и хранится в том же каталоге, где находятся файлы конфигурации, в файле с именем `iplir.db-<интерфейс>`. Максимальный размер журнала устанавливается пользователем.

Записи о пакетах накапливаются в журнале до тех пор, пока не будет достигнут максимальный размер журнала, после чего самые старые записи стираются и на их место записываются новые. Для уменьшения объема журнала и удобства его просмотра одинаковые записи о пакетах, зарегистрированные в течение короткого времени, объединяются в одну запись, и затем при просмотре журнала можно узнать, сколько раз произошло событие, описываемое этой записью.

Секция [db] содержит следующие параметры:

- `maxsize` – максимальный размер журнала в мегабайтах (1 мегабайт = 1048576 байт).



Внимание! Для ПАК ViPNet Coordinator HW100 максимальный размер журнала должен быть не более 10МБ. При указании большего размера он принудительно устанавливается в 10МБ.

Реальный размер журнала из-за наличия в нем служебного заголовка получается примерно на 1 Кбайт больше. При старте управляющего демона после размера журнала автоматически дописывается слово `Mbytes`, чтобы облегчить понимание значения этого параметра. При последующем редактировании это слово можно оставить, а можно стереть – оно будет добавлено автоматически. Значение параметра 0 отключает ведение журнала. Если до отключения журнала в нем были записи, то они не удаляются, но просмотреть их нельзя.

- `timediff` – интервал времени, в течение которого одинаковые события объединяются в журнале в одну запись. Задается в секундах, значение по умолчанию 60 (секунд).

В случае задания нулевого значения объединение событий отключается. Если объединение событий для журнала выключено, то при очень интенсивном трафике не все пакеты могут регистрироваться в журнале.

- `registerall` – включение/отключение регистрации всех пакетов, проходящих через данный интерфейс. Может принимать значение `on` или `off`, значение по умолчанию `off`. Если параметр `registerall` установлен в значение `off`, то регистрируются только блокированные пакеты, а также события об изменении адресов сетевых узлов.

- `registerbroadcast` – включение/отключение регистрации широковещательных пакетов. Может принимать значение `on` или `off`, значение по умолчанию `off`.
- `registertcpserverport` – включение/отключение регистрации порта клиента при соединении ТСР. Может принимать значение `on` или `off`, значение по умолчанию `off`.

Как правило, порт клиента при ТСР-соединении выделяется динамически и никакой полезной информации не несет. Если с какого-либо сетевого ресурса производятся попытки подсоединиться к какому-либо порту на компьютере, а соединение по каким-то причинам не будет установлено, то при следующей попытке установить соединение с того же ресурса будет использоваться другой номер порта. При использовании сканеров портов или каких-либо сетевых атак число таких попыток может достигать нескольких сотен в секунду. Поскольку клиент использует каждый раз разные порты, то такие пакеты не считаются одинаковыми и для каждого из них создается своя запись в журнале, что засоряет его и затрудняет последующий анализ. При установке параметра `registertcpserverport` в значение `off` порт клиента при ТСР-соединении не регистрируется и не учитывается, что позволяет объединить события о попытках присоединиться к какому-либо порту на компьютере с определенного адреса в одну запись, что часто бывает очень удобно.

Работа с политиками безопасности

Политика безопасности формируется в Центре управления политиками безопасности (ЦУПБ) и рассылается на сетевые узлы с помощью транспортного модуля MFTR. Она определяет текущую политику безопасности на узле совместно с правилами, заданными пользователем на самом узле (в файлах конфигурации).

Политика безопасности включает в себя предправила и постправила фильтрации открытых IP-пакетов (локальных, транзитных и широковещательных) и правила задания режимов безопасности на интерфейсах. Предправила предшествуют правилам, заданным пользователем на узле, постправила следуют после пользовательских правил. Режимы безопасности на интерфейсах, установленные политикой безопасности, имеют приоритет над режимами, заданными пользователем на узле.

В правилах политики безопасности вместо конкретных значений может быть указана специальная лексическая единица, называемая подстановкой. Подстановка задает некоторое условие и результат, который вставляется в правило вместо подстановки при выполнении условия. Подробное описание синтаксиса правил политики безопасности и формата подстановок содержится в документе «ViPNet Policy Manager. Руководство администратора».

Обработка политики безопасности производится управляющим демоном каждый раз при его старте/рестарте. В результате обработки правила, содержащиеся в полученной из ЦУПБ политике безопасности, добавляются в файлы конфигурации `firewall.conf` и `iplir.conf-<интерфейс>` в виде следующих параметров:

- предправила – параметры `policy-pre-rule`;
- постправила – параметры `policy-post-rule`;
- режимы безопасности на интерфейсах – параметры `policy-mode`.



Примечание. Не следует редактировать указанные параметры вручную, так как после старта/рестарта управляющего демона они будут заменены правилами из политики безопасности.

Перед обработкой политики безопасности из файлов конфигурации удаляются все правила, добавленные при предыдущей обработке политики безопасности:

- из файла `firewall.conf` удаляются все параметры `policy-pre-rule` и `policy-post-rule` из всех секций, где они присутствуют;
- из всех файлов `iplir.conf-<интерфейс>`, где в секции `[mode]` присутствуют параметры `policy-mode`, эти параметры удаляются.

Обработка политики безопасности осуществляется в следующей последовательности:

- 1 Обработываются и раскрываются подстановки (если они присутствуют в политике безопасности). При обнаружении ошибок в формате подстановок сообщение об этом выдается в `syslog` и обработка политики безопасности прекращается.
- 2 Проверяется синтаксическая корректность правил, полученных после обработки подстановок. При обнаружении хотя бы одного некорректного правила сообщение об этом выдается в `syslog` и обработка политики безопасности прекращается.
- 3 В секции `[local]`, `[forward]` и `[broadcast]` файла `firewall.conf` добавляются параметры `policy-pre-rule` и `policy-post-rule` согласно политике безопасности. Параметры `policy-pre-rule` располагаются в начале каждой секции в порядке их следования в политике безопасности, параметры `policy-post-rule` – в конце каждой секции в порядке их следования в политике безопасности.
- 4 Определяются интерфейсы, для которых должен быть установлен режим безопасности согласно политике безопасности. Для этих интерфейсов в файлы `iplir.conf-<интерфейс>` в секцию `[mode]` добавляется параметр `policy-mode` с соответствующим значением режима.

Настройка режимов работы через межсетевой экран

Если ПАК ViPNet Coordinator HW работает во внутренней сети с внутренними IP-адресами, и на входе этой сети установлен межсетевой экран или другое устройство, осуществляющее преобразование внутренних адресов в адреса, доступные из внешней сети (то есть выполняется NAT – Network Address Translation), то для нормальной работы с другими узлами, находящимися не во внутренней сети или стоящими за другими межсетевыми экранами, необходимо настроить один из режимов работы через межсетевой экран (см. «[Режимы работы ПО ViPNet через межсетевой экран](#)» на стр. 23). При настройке режима необходимо помнить, что перед редактированием файла конфигурации `iplir.conf` необходимо остановить управляющий демон, а после окончания редактирования вновь запустить его, чтобы все изменения вступили в действие.

Настройка режима «Без использования межсетевого экрана»

При выборе данного режима необходимо установить следующие параметры в файле `iplir.conf`:

- В секции `[id]`, описывающей ПАК, установить параметр `usefirewall` в значение `off`, т.е. внешний межсетевой экран не используется.
- В секции `[dynamic]` установить параметр `dynamic_proxy` в значение `off`.
- Для всех используемых сетевых интерфейсов в секциях `[adapter]` установить параметр `type` в значение `internal`.

Настройка режима «Координатор»

ПАК может устанавливаться за Координатор, только если один из сетевых интерфейсов этого Координатора доступен ПАК по реальному адресу (то есть между ними нет никаких межсетевых экранов).

Для настройки работы ПАК через Координатор, выбранный в качестве прокси-сервера, необходимо установить следующие параметры в файле `iplir.conf`:

- В секции `[id]`, описывающей Координатор, задать значение любого из его реальных IP-адресов, доступных ПАК (параметр `ip`), если он еще не задан; в этой же секции задать значение параметра `port`, т.е. порта назначения, на который будут посылааться пакеты для данного Координатора.
- В секции `[adapter]`, соответствующей сетевому интерфейсу, со стороны которого установлен Координатор, установить значение параметра `type` в `external`.
- В секции `[id]`, описывающей ПАК, параметр `usefirewall` выставить в значение `on`; в этой же секции значение параметра `proxyid` выставить равным значению `id` Координатора.
- В секции `[dynamic]` параметр `dynamic_proxy` выставить в значение `off`.

После соединения с Координатором и если он правильно настроен, в секции `[id]`, описывающей Координатор, установятся значения `firewallip`, `port` и `proxyid`. Кроме того, могут измениться значения параметров `firewallip` и `port` в секции `[id]`, описывающей ПАК – они должны соответствовать параметрам выбранного Координатора.

Настройка режима «Со статической трансляцией адресов»

Если ПАК установлен за межсетевым экраном (устройством) с трансляцией адресов (NAT), на котором можно настроить статические правила NAT, то на ПАК нужно произвести настройки режима работы со статическим NAT.

В этом случае IP-адрес ПАК и порт доступа к нему должны быть жестко заданы на межсетевом экране. Кроме того, на ПАК необходимо настроить маршрутизацию на внешний межсетевой экран (шлюз по умолчанию или маршруты для удаленных подсетей).

Для пропуска пакетов на межсетевом экране (или NAT-устройстве), на котором можно настроить статические правила NAT, должен быть обеспечен:

- Пропуск исходящих UDP-пакетов с IP-адресом и портом ПАК (Source-порт) на любой внешний адрес и порт (с подменой адреса источника на внешний адрес NAT-устройства).

- Пропуск и перенаправление входящих UDP-пакетов с портом ПАК (Destination-порт) на IP-адрес ПАК.

Для настройки работы ПАК через межсетевой экран со статическим NAT необходимо установить следующие параметры в файле `iplir.conf`:

- В собственной секции `[id]` установить параметр `usefirewall` в значение `on`.
- В секции `[dynamic]` установить параметр `dynamic_proxy` в значение `off`.
- В секции `[adapter]`, соответствующей сетевому интерфейсу, со стороны которого установлен внешний межсетевой экран, установить параметр `type` в значение `external`.
- В собственной секции `[id]` установить параметр `proxyid` в значение `0`; при необходимости значение параметра `port` выбрать из диапазона `1-65535` (по умолчанию `55777`). Этот параметр определяет номер порта, от которого преобразованные в UDP-формат пакеты уходят с ПАК (Source-порт), и на который такие пакеты приходят на ПАК (Destination-порт). Номер порта имеет смысл изменять, только если внутри локальной сети через один межсетевой экран (или NAT-устройство) работает несколько узлов с ПО ViPNet. В этом случае у всех таких узлов номера портов должны быть разные.

В случае если на внешнем межсетевом экране с NAT есть возможность настроить статические правила только для входящих пакетов, предназначенных ПАК, то есть обеспечить пропуск пакетов, имеющих заданный адрес и порт назначения, а также их перенаправление на адрес ПАК, то следует установить параметр `fixfirewall` в значение `on`. В этом случае IP-адрес ПАК и порт доступа к нему должны быть жестко заданы на межсетевом экране, и их необходимо прописать в параметрах `firewallip` и `port` в собственной секции `[id]`.

В случае если внешний адрес межсетевого экрана с NAT в процессе работы может меняться, то следует установить параметр `fixfirewall` в значение `off`. В этом случае на других узлах IP-адрес доступа к ПАК будет регистрироваться по внешним параметрам пакета. Параметр `firewallip` в данном режиме определяется автоматически по информации, полученной от узлов, находящихся во внешней сети, поэтому редактировать его вручную не следует.

Информация об IP-адресе межсетевого экрана и порте доступа сообщается программой всем остальным узлам, с которыми связан ПАК.

Настройка режима «С динамической трансляцией адресов»

Если в локальной сети подключение к сети происходит через некоторое устройство, выполняющее NAT, на котором затруднительно настроить статические правила трансляции, и есть необходимость во взаимодействии с другими узлами, находящимися во внешней относительно этого устройства сети, то на узле нужно настроить режим работы с динамической трансляцией адресов.

Режим работы с использованием такого типа межсетевого экрана наиболее универсален и может использоваться практически во всех случаях. Однако основное его назначение – обеспечить надежное двухстороннее соединение с узлами, работающими через NAT-устройства, на которых настройка статических правил трансляции затруднена или невозможна. Такая ситуация типична для простых NAT-устройств с минимумом настроек, например, DSL-модемов, Wireless-устройств и в других случаях. Затруднительно также произвести настройки на NAT-устройствах, установленных у провайдера (в домашних сетях Home network, GPRS и других сетях, где провайдер предоставляет частный IP-адрес).

Далее кратко описывается технология пропуска трафика такими NAT-устройствами. Все NAT-устройства обеспечивают пропуск UDP-трафика в режиме автоматического создания так называемых динамических правил NAT, когда пропускаются любые исходящие пакеты с подменой адреса и порта, фиксируются их параметры, и на основании этих параметров создаются динамические правила пропуска для входящего трафика. В течение определенного промежутка времени (тайм-аута) пропускаются входящие пакеты, параметры которых соответствуют созданным правилам. По истечении тайм-аута после прохождения последнего исходящего пакета соответствующие динамические правила удаляются, и входящие пакеты начинают блокироваться. То есть инициативное соединение извне с узлами, работающими через такие NAT-устройства, невозможно, если не будет соответствующего исходящего трафика.

Для обеспечения возможности двухсторонней работы на узле, работающем через NAT-устройство, и всех его Клиентов (если узел является Координатором) на узле устанавливается режим работы через межсетевой экран с динамической трансляцией адресов. Одновременно должен присутствовать постоянно доступный Координатор, находящийся во внешней сети (см. рисунок ниже). Назовем его Координатором входящих соединений (параметр `forward_id` секции `[dynamic]`). Узел в режиме использования типа межсетевого экрана с динамическим NAT после подключения к сети производит с заданным периодом (по умолчанию – 25 секунд) отправку UDP-пакетов на свой Координатор входящих соединений (параметр `timeout` секции `[dynamic]`). Период отправки этих пакетов не должен намного превышать тайм-аут сохранности динамического правила на NAT-устройстве. У разных NAT-устройств этот тайм-аут устанавливается разный, но обычно не менее 30 секунд. Это позволяет любому внешнему узлу в любой момент прислать на данный узел через его Координатор входящих

соединений пакет любого типа. Исходящие пакеты в рассматриваемом режиме узел всегда отправляет напрямую адресату, минуя свой Координатор входящих соединений. После первого полученного в ответ пакета внешний узел автоматически начинает передавать весь трафик напрямую данному узлу, работающему через устройство с NAT (см. рисунок ниже). Такая технология обеспечивает постоянную доступность узла, работающего через NAT-устройство (т.к. на NAT-устройстве не удаляются динамические правила), и одновременно существенно увеличивает скорость обмена между узлами.

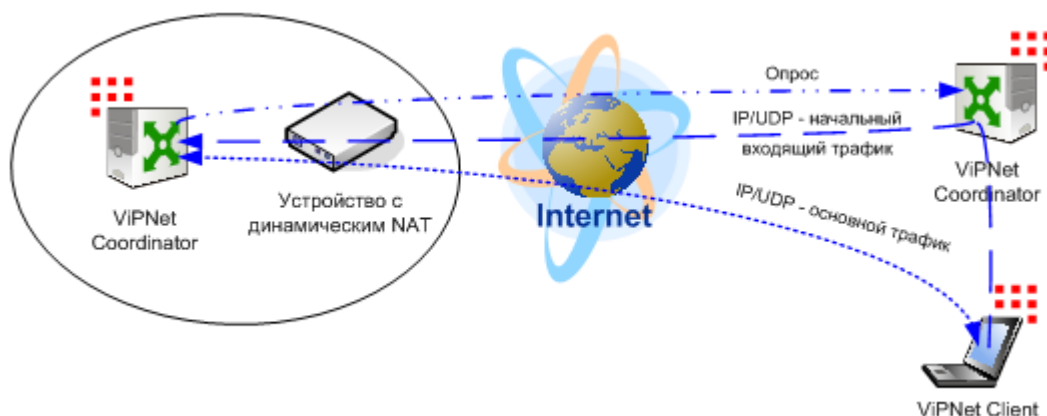


Рисунок 31: Организация трафика узла, работающего через устройство с динамическим NAT

Если все же есть проблемы со связью в режиме межсетевого экрана с динамическим NAT, то существует возможность включить опцию (параметр `always_use_server` секции `[dynamic]`), которая позволяет направлять любой трафик с внешними узлами через Координатор. Если включить эту опцию, то все соединения с другими узлами будут происходить только через выбранный Координатор для организации входящих соединений, т.е. технология, описанная выше, использоваться не будет. В этом режиме из-за удлинения маршрута прохождения пакетов возможно снижение скорости обмена.



Примечание. Узел в режиме работы с динамической трансляцией адресов для взаимодействия с узлами, работающими через какой-либо Координатор, должен быть связан с этим Координатором.

Для настройки работы ПАК в данном режиме необходимо установить следующие параметры в файле `iplir.conf`:

- В собственной секции `[id]` установить параметр `usefirewall` в значение `on`.
- В собственной секции `[id]` значение параметра `port` выбрать из диапазона 1-65535 (обычно 55777), если оно еще не установлено.

- В секции `[adapter]`, соответствующей сетевому интерфейсу, со стороны которого установлен внешний межсетевой экран, установить параметр `type` в значение `external`.
- В секции `[dynamic]` установить параметр `dynamic_proxy` в значение `on`.
- В секции `[dynamic]` выбрать внешний Координатор для организации входящих соединений – параметр `forward_id`, если он еще не выбран. Данный параметр выставляется вручную, но не может принимать нулевое значение.



Примечание. Выбранный Координатор должен иметь хотя бы один адрес доступа, иначе включение рассматриваемого режима будет невозможно.

- При необходимости изменить значения параметров `timeout` и `always_use_server` в секции `[dynamic]`.

Параметры `firewallip` и `port` секции `[dynamic]` в данном режиме определяются автоматически по информации, полученной от узлов, находящихся во внешней сети, поэтому не следует редактировать эти параметры вручную.

Настройка работы с удаленным Координатором через фиксированный альтернативный канал

Согласно базовой логике работы, каждый узел сети ViPNet взаимодействует с каждым другим узлом по определенному адресу доступа и порту, которые задаются параметрами `firewallip` и `port` в секции `[id]`. При изменении адреса доступа какого-либо узла эта информация рассылается всем остальным узлам сети ViPNet с помощью механизма служебных рассылок. Эта схема хорошо работает, когда у каждого узла есть только один возможный адрес доступа в каждый момент. Однако нередко некоторые узлы подключены к нескольким независимым каналам связи и имеют несколько адресов доступа, каждый из которых может использоваться для связи с другими узлами. При этом отсутствие контроля над выбором канала зачастую приводит к проблемам, связанным с тем, что узлы могут периодически переключаться между каналами, использовать более «дорогой» канал и т.д.

Для управления выбором каналов связи между Координаторами ПО ViPNet предоставляет специальный механизм, позволяющий пользователю принудительно установить нужный канал для обмена трафиком. Для этого в файле конфигурации `iplir.conf` существует секция `[channels]`, в которой задается набор возможных каналов связи с возможной регистрацией групп узлов (Координаторов) в определенных каналах. При определении имени канала в секции `[channels]` не обязательно регистрировать в этом канале какие-либо группы. В этом случае будет считаться, что для работы через данный канал не зарегистрированы никакие группы узлов. Если группы узлов заданы, это означает, что они зарегистрированы для работы через этот канал. Однако это не означает, что любой узел группы переключится на работу через этот канал, так как необходимо еще задать внешний IP-адрес доступа и порт доступа (опционально) для этого канала. Эти значения задаются для каждого узла группы параметрами `channelfirewallip` и `channelport` соответственно (подробное описание синтаксиса этих параметров см. [Секция \[id\]](#) (на стр. 90)). То есть для двух различных узлов одной и той же группы значения `channelfirewallip` и `channelport` для одного и того же канала могут быть разными. Это позволяет производить гибкую настройку в ряде случаев, например, когда для доступа по одному и тому же каналу к различным узлам используются различные порты доступа.

Таким образом, каждый канал характеризуется именем (уникальный идентификатор), зарегистрированными в нем группами узлов и параметрами доступа, индивидуально настраиваемыми для каждого узла группы. В целом логику выбора канала для удаленного Координатора **К** можно описать следующим образом:

- если в секции `[channels]` определен канал **А** и в нем зарегистрирована группа **В** (`channel= A, B`),
- если в группу **В** входит удаленный Координатор **К** (в секции `[id]` для этого Координатора задан соответствующий параметр `group= B`),
- если в секции `[id]` Координатора **К** задано значение `channelfirewallip` для этого же канала (`channelfirewallip= A, IP`),

то работа с Координатором **К** будет всегда осуществляться через канал **А** с внешним адресом доступа `IP`.

Если при этом в секции `[id]` для Координатора **К** помимо параметра `channelfirewallip` присутствует параметр `channelport` для выбранного канала **А** (`channelport= A, Port`), то в качестве порта доступа будет всегда использоваться значение `Port`.

Важно помнить, что для того чтобы данный узел всегда работал с удаленным Координатором через выбранный канал, необходимо на самом удаленном Координаторе выполнить симметричные настройки, настроив тот же самый канал для взаимодействия с данным узлом. В противном случае может возникнуть ситуация, когда исходящие пакеты с данного узла будут направляться на удаленный Координатор по выбранному каналу, а пакеты с другой стороны будут приходить по другому каналу.

В случае выбора для работы с удаленным Координатором фиксированного канала с помощью описанного выше способа, значения параметров `firewallip` и `port` (если задан параметр `channelport`) в секции `[id]` Координатора будут автоматически заменены после старта управляющего демона значениями параметров `channelfirewallip` и `channelport` соответственно. При этом все пакеты в адрес данного Координатора будут отправляться через выбранный канал на адрес доступа `channelfirewallip` и порт `channelport`, независимо от того, существует ли физическое соединение по этому каналу или нет. Автоматического перехода на другой канал не произойдет.

Для определения другого канала взаимодействия с удаленным Координатором необходимо задать параметры `channelfirewallip` и `channelport` для другого канала в секции `[id]` Координатора, а также определить этот канал в секции `[channels]`.

Переключение между каналами для Координатора может быть осуществлено двумя способами:

- 1 Перерегистрацией Координатора в другой группе, которая зарегистрирована для работы через нужный канал, т.е. изменением значения параметра `group` в секции `[id]` Координатора. Данный способ наиболее предпочтителен в случае, если требуется переключить на другой канал только один узел, при этом другие узлы группы остаются работать через старый канал.
- 2 Перерегистрацией группы, в которую входит Координатор, для работы через другой канал, т.е. переносом имени группы в качестве значения другого параметра `channel` секции `[channels]`. Данный способ более предпочтителен в случае переключения группы узлов для работы через другой канал, так как в этом случае переключение канала произойдет для всех узлов группы, у которых задан параметр `channelfirewallip` для соответствующего канала. Кроме того, этот способ удобен для случая, когда в группу входит только один узел.

Отключение механизма работы через фиксированный канал может быть осуществлено следующими способами:

- 1 В секции `[id]` для соответствующего Координатора удалить параметр `group`. При этом узел не будет зарегистрирован в какой-либо группе. Этот способ удобен, когда необходимо временно отключить данный механизм, так как для его повторного включения нужно будет лишь снова задать параметр `group` для этого узла.
- 2 В секции `[channel]` для соответствующего канала удалить из списка групп нужную группу. При этом работа через фиксированный канал будет отключена для всех узлов группы.

Примеры настроек для выбора, переключения и отключения альтернативных каналов приведены в Приложении (см. «[Примеры настроек работы ПАК через фиксированные альтернативные каналы](#)» на стр. 241).



Внимание! Описанный в данном разделе механизм рассчитан на использование в схемах, в которых на сетевых каналах между Координаторами либо не применяются NAT-устройства, либо применяются NAT-устройства со статической трансляцией адресов. При этом NAT-устройство не должно изменять в процессе работы свой внешний адрес, а также порт отправителя.

В случае использования NAT-устройств с динамической трансляцией адресов работоспособность механизма установки фиксированного канала связи не гарантируется!

Настройка ПАК, выполняющего функции Сервера Открытого Интернета

Если в организации из соображений политики безопасности компьютерам локальной сети запрещен выход в Интернет (назовем их группой компьютеров А), но отдельным компьютерам необходим такой доступ (группа компьютеров Б), то для группы компьютеров Б можно с использованием технологии ViPNet создать в локальной сети выделенный виртуальный контур компьютеров, трафик которых при работе в сети Интернет был бы полностью изолирован от остальной локальной сети.

Для решения данной задачи на границе сети устанавливается Координатор (ПАК) с функцией Сервера Открытого Интернета. Одновременно на компьютере, который туннелируется ПАК и расположен за отдельным интерфейсом в демилитаризованной зоне, устанавливается любой прокси-сервер прикладного уровня типа Squid. На компьютеры группы Б устанавливается ПО ViPNet Client, и эти сетевые узлы связываются в ЦУСе с ПАК.

Возможны два способа организации работы:

- 1 Компьютерам группы Б разрешается работа исключительно в Интернете.

В этом случае сетевым узлам в ЦУСе не предоставляются никакие другие связи, кроме связи с Сервером Открытого Интернета (Координатором). ПО ViPNet устанавливается в первый режим безопасности (см. «[Основные режимы безопасности ПО ViPNet](#)» на стр. 21). Таким образом трафик с Интернетом, проходя через прокси-сервер прикладного уровня, на участке между компьютером группы Б и Координатором (то есть в локальной сети) будет упакован в VPN-соединение. Такой трафик при любых атаках не может выйти за пределы созданного виртуального контура. Данное решение эквивалентно физическому выделению сегмента локальной сети для работы в Интернете.

- 2 Компьютерам группы Б требуется также предоставить возможность работы с другими защищенными узлами и открытыми ресурсами локальной сети.

Этот способ менее безопасен, чем первый. Тем не менее он обеспечивает достаточно высокий уровень защиты от атак как компьютеров группы Б, так и компьютеров группы А. Цель достигается путем разделения времени работы в сети Интернет и в локальной сети. При этом на компьютере группы Б:

- при работе в Интернете блокируется любой трафик как в локальную сеть, так и с другими сетевыми узлами, кроме Сервера Открытого Интернета;
- при работе в локальной сети блокируется любой трафик с Интернетом.

Такое разделение во времени исключает любые онлайн-атаки на компьютеры группы А через компьютеры группы Б.

В качестве Сервера Открытого Интернета Координатор выполняет следующие функции:

- организует доступ к ресурсам открытого Интернета через прокси-сервер прикладного уровня;
- запрещает доступ к ресурсам открытого Интернета для компьютеров группы А;
- организует защищенный туннель между собой и компьютером группы Б на время его работы с ресурсами открытого Интернета, без возможности доступа к этому туннелю со стороны всех остальных пользователей локальной сети;
- является межсетевым экраном, который запрещает доступ в локальную сеть из открытого Интернета.

Технология гарантирует, что никакие стратегии атак как снаружи, так и изнутри сети не могут привести к нарушению безопасности компьютеров сети, подключение которых к Интернету запрещено (компьютеры группы А). На такие компьютеры трафик из Интернета при любых атаках может попасть только в зашифрованном виде, что не будет воспринято компьютером и в связи с этим не опасно. Попытки проведения атак на сеть через компьютеры группы Б невозможны, так как блокируется посторонний трафик от них, кроме трафика с Сервером Открытого Интернета. Одновременно исключаются любые возможности несанкционированного подключения из локальной сети к Интернету.

Для настройки ПАК, выполняющего функции Сервера Открытого Интернета, необходимо:

- В файле конфигурации для сетевого интерфейса со стороны локальной сети с помощью параметра `mode` установить первый режим безопасности (в этом режиме блокируется любой открытый трафик, как снаружи, так и изнутри локальной сети).
- В файле конфигурации для сетевого интерфейса со стороны Интернета установить режим безопасности 2.
- В секцию `[local]` файла `firewall.conf` (см. «[Настройка правил обработки открытых IP-пакетов](#)» на стр. 112) добавить два правила:

- правило, разрешающее одностороннее соединение от всех сетевых узлов локальной сети в сторону IP-адреса туннелируемого прокси-сервера;
- правило, разрешающее одностороннее соединение от прокси-сервера в сторону всех адресов за интерфейсом со стороны Интернета.

Компьютеры, для которых организуется выход в Интернет, могут взаимодействовать с ПАК только через VPN-соединения и только по служебным протоколам, которые контролируются ПО ViPNet. Таким образом обеспечивается полная безопасность ПАК.



Настройка конфигурации транспортного модуля

Назначение и функциональность транспортного модуля	149
Настройка параметров транспортного модуля	151

Назначение и функциональность транспортного модуля

Транспортный модуль предназначен для обеспечения надежной и безопасной передачи транспортных конвертов между узлами сети ViPNet посредством протоколов TCP (этот канал передачи называется MFTR) и SMTP/POP3. Кроме того, транспортный модуль принимает непосредственное участие в удаленном обновлении адресных справочников и/или ключевых баз на узлах, а также в удаленном обновлении ПО ViPNet.

Функциональность транспортного модуля в составе ПАК ViPNet Coordinator HW100 зависит от лицензионных ограничений на ПАК (см. «[Лицензионные ограничения ПАК ViPNet Coordinator HW100](#)» на стр. 39). При наличии лицензии, не предусматривающей функцию Сервера-маршрутизатора, транспортный модуль выполняет только функции, связанные с удаленным обновлением на ПАК собственных адресных справочников, ключевых баз и ПО ViPNet: транспортный модуль принимает только конверты, содержащие адресованные ПАК обновления, все прочие конверты не принимаются.



Внимание! При формировании в ЦУСе сети ViPNet **нельзя регистрировать Клиентов на ПАК ViPNet Coordinator HW100 с лицензией без поддержки функции Сервера-маршрутизатора.**

Чтобы Клиенты, установленные в сети за ПАК ViPNet Coordinator HW100 с лицензией без поддержки функции Сервера-маршрутизатора, могли получать транспортные конверты, их следует регистрировать на том Координаторе, который выполняет функцию Сервера-маршрутизатора. При наличии лицензии с поддержкой функции Сервера-маршрутизатора транспортный модуль обладает полной функциональностью, и на таком ПАК можно регистрировать Клиентов.

Транспортный модуль в составе ПАК ViPNet Coordinator HW1000 и HW-VPNМ не имеет ограничений по функциональности, и на этих ПАК можно регистрировать Клиентов.

Транспортные конверты представляют собой конверты с данными, которыми обмениваются между собой приложения, входящие в состав ПО ViPNet. Транспортный модуль передает конверты в соответствии с адресами получателей, прописанными в заголовках этих конвертов.

При связи по **каналу MFTP** устанавливается соединение TCP с узлом-получателем конвертов, проводится взаимная аутентификация узлов и осуществляется прием/передача конвертов друг для друга.

При связи по **каналу SMTP/POP3** транспортный модуль переадресует конверты для отправки модулю MailTrans (см. «Секция [mailtrans]» на стр. 155), который передает их через сервер SMTP, а также забирает с сервера POP3 конверты, предназначенные для этого узла.

Все настройки транспортного модуля содержатся в его конфигурационном файле, который называется `mftp.conf`. Для его редактирования используется команда `mftp config` (см. «Команды группы `mftp`» на стр. 79).



Примечание. Для правильной работы транспортного модуля необходимо внести соответствующие изменения в его конфигурационный файл.

Настройка параметров транспортного модуля

Конфигурационный файл MFTP состоит из секций, каждая из которых содержит ряд параметров. Имена секций заключены в прямые скобки, например `[channel]`, `[misc]`. Значения параметров отделяются от их идентификаторов знаком «=» и следующим за ним пробелом, например: `ip= 192.168.201.1`.

Секция `[channel]`

Секции `[channel]` содержат настройки каналов, по которым ПАК может осуществлять обмен с другими сетевыми узлами. Каждому каналу соответствует своя секция `[channel]`. Количество параметров в каждой секции зависит от типа выбранного канала. По умолчанию при создании файла конфигурации все типы каналов устанавливаются в значение `mftp`.



Внимание! Добавление и удаление секций `[channel]` осуществляется автоматически, поэтому не следует добавлять и удалять секции данного типа вручную!

Секции `[channel]` содержат ряд общих параметров для каналов любого типа:

- `id` – уникальный 4-х байтовый идентификатор сетевого узла, с которым устанавливается обмен по данному каналу. Идентификатор представлен в шестнадцатеричном виде, например: `id= 0x270e000a`.



Внимание! Менять параметр `id` вручную не следует!

- `name` – имя сетевого узла. Этот параметр носит информационный характер.
- `type` – тип канала. Может принимать следующие значения: `mftp`, `smtp`. По умолчанию значение параметра `mftp` для всех узлов. Если данный параметр отсутствует, то используется значение по умолчанию.

- `off_flag` – признак отключения канала (`yes/no`). По умолчанию значение параметра `no`. Установка параметра в значение `yes` позволяет временно отключить канал. В таком случае исходящие конверты, передаваемые по этому каналу, будут оставаться в очереди до тех пор, пока канал не будет включен или инициатором соединения по данному каналу не станет удаленный Сервер-маршрутизатор (Координатор). Если инициатором соединения в данном случае станет удаленный Клиент, то предназначенные ему конверты не отправляются, а этому Клиенту передается специальная команда, которая выключает соответствующий канал в настройках его транспортного модуля. Если данный параметр отсутствует, то используется значение по умолчанию.
- `call_flag` – признак немедленной передачи конвертов по каналам MFTR и SMTP (`yes/no`). По умолчанию значение параметра `yes`. При установке значения параметра в `yes` попытка передачи конверта по данному каналу будет производиться немедленно. В противном случае конверт будет оставаться в очереди до тех пор, пока инициатором соединения по данному каналу не станет удаленный узел (в случае MFTR-канала) или не будет вызван модуль MailTrans (в случае SMTP-канала). Если данный параметр отсутствует, то используется значение по умолчанию.

Для каждого из типов каналов существуют специфические параметры.

Специфические параметры для канала MFTR

Для канала MFTR в секции `[channel]` дополнительно задаются следующие параметры:

- `ip` – IP-адрес удаленного сетевого узла. Значение данного параметра запрашивается у управляющего демона. Если оно по каким-либо причинам не было сообщено (равно 0.0.0.0), то его можно задать вручную, а затем перезапустить транспортный модуль. Данный параметр может изменяться в процессе работы, поэтому корректировать его вручную не рекомендуется.
- `call_timeout` – интервал опроса (в секундах) соответствующего удаленного сетевого узла. Время следующего опроса отсчитывается от момента разрыва последнего соединения с этим узлом.
При значении параметра `-1` опрос не производится. По умолчанию значение параметра `-1` для всех узлов. Если данный параметр отсутствует, то используется значение по умолчанию.
- `last_port` – значение порта, по которому осуществлялось последнее удачное MFTR-соединение. Это значение будет использоваться при следующей попытке соединения с этим узлом.



Внимание! Менять параметр `last_port` вручную не следует!

- `last_call` – время последней попытки опроса данного канала.
-



Внимание! Менять параметр `last_call` вручную не следует!

- `last_err` – время, когда произошла последняя ошибка при попытке соединения или в процессе передачи данных.
-



Внимание! Менять параметр `last_err` вручную не следует!

Специфические параметры для канала SMTP

Для канала SMTP в секции `[channel]` дополнительно задаются следующие параметры:

- `reportaddress` – адрес ящика электронной почты, в который будут отправляться исходящие конверты. Адрес задается в соответствии со следующим правилом:

```
reportaddress= <username>@<servername>.<domain>
```
- `version` – версия протокола инкапсуляции конверта MFTP в почтовый конверт RFC-822, передаваемый по каналу SMTP. Возможные значения параметра: 1.0 и 2.0. Данный параметр определяет версию протокола для конкретного канала. В случае отсутствия параметра в секции используется протокол, определяемый глобально для всех каналов SMTP в секции `[mailtrans]` (см. «Секция [\[mailtrans\]](#)» на стр. 155).
- `maxsize` – максимальный размер (в килобайтах) почтового SMTP-конверта при отправке. Используется в случае, если установлена версия протокола 2.0 (см. параметр `version`). При отправке MFTP-конверт разбивается на несколько SMTP-конвертов, размер каждого из которых не превышает заданный параметром `maxsize`. Значение 0 означает, что ограничение на размер SMTP-конвертов отсутствует. Допустимые значения данного параметра: от 100 до 2048000 (2 ГБ). В случае отсутствия параметра в секции используется значение `maxsize`, определяемое глобально для всех каналов SMTP в секции `[mailtrans]` (см. «Секция [\[mailtrans\]](#)» на стр. 155).

Секция [transport]

Данная секция содержит ряд параметров, определяющих пути к транспортным каталогам, то есть к каталогам, участвующим в обмене конвертами, их обработке и т.п. Эти параметры задают лишь основные каталоги. Вспомогательные каталоги создаются транспортным модулем в процессе работы как подкаталоги основных. При первом создании конфигурационного файла значения параметров этой секции определены по умолчанию относительно каталога ключевых баз.



Примечание. Транспортный модуль при каждом запуске проверяет существование каталогов, заданных этими параметрами, и при необходимости создает их.

Секция [transport] содержит следующие параметры:

- `in_path` – полный путь к каталогу, в который помещаются полностью принятые конверты. По умолчанию значение параметра `basedir/in`, где `basedir` – полный путь к каталогу ключевых баз.
- `out_path` – полный путь к каталогу, в который внешние приложения помещают сформированные конверты для отправки. По умолчанию значение параметра `basedir/out`.
- `trash_path` – полный путь к каталогу, в который помещаются устаревшие конверты из исходящей очереди – так называемая «корзина». По умолчанию значение параметра `basedir/trash`.

Секция [upgrade]

В данной секции присутствуют параметры, которые определяют поведение транспортного модуля при приеме обновления. Секция [upgrade] содержит следующие параметры:

- `upgrade_path` – полный путь к каталогу, в который помещаются файлы обновления после распаковки соответствующих конвертов. По умолчанию значение параметра `basedir/ccc`, где `basedir` – полный путь к каталогу ключевых баз.
- `upgrade_ini` – имя файла конфигурации для процесса обновления. По умолчанию значение параметра `basedir/user/upgrade.conf`.

- `upgrade_for_kc_path` – полный путь к каталогу, в который внешние приложения помещают файлы с запросами сертификатов `*.sok`. По умолчанию значение параметра `basedir/ccc/for_kc`.
- `upgrade_checktimeout` – интервал (в секундах) периодической проверки транспортного каталога, заданного параметром `upgrade_path`, на наличие файлов обновления. В случае соответствия файлов обновления условиям обновления (время обновления и т.д.) происходит вызов модуля обновления. По умолчанию значение параметра 300 (секунд).
- `confsave` – тип сохраняемой конфигурации при автосохранении перед проведением обновления (см. «Работа с конфигурациями ViPNet» на стр. 175). Может принимать следующие значения: `full`, `partial` и `off`. При значении параметра `full` производится автосохранение полной конфигурации, при значении `partial` – частичной конфигурации. При значении параметра `off` автосохранение не производится. По умолчанию значение параметра `partial`.
- `maxautosaves` – количество автосохраненных конфигураций в базе. Перед очередным автосохранением конфигурации проверяется число автосохраненных конфигураций. Если это число больше или равно значению `maxautosaves`, то из базы удаляются самые старые автосохраненные конфигурации в таком количестве, чтобы осталось `maxautosaves-1`, после чего сохраняется текущая конфигурация. При этом в системный журнал `syslog` выводится соответствующее сообщение. Текущая версия транспортного модуля имеет ограничение на максимальное количество автосохраненных конфигураций – не более 10, поэтому значение параметра `maxautosaves` не может превышать 10. При попытке установки большего значения оно принудительно устанавливается равным максимально допустимому значению.

Секция [mailtrans]

В данной секции присутствуют параметры, которые определяют взаимодействие транспортного модуля с модулем почтового обмена MailTrans. Секция [mailtrans] содержит следующие параметры:

- `mailtrans_bin` – полный путь к исполняемому файлу модуля почтового обмена. По умолчанию значение параметра `/sbin/mailtrans`.
- `inputmailbox` – адрес почтового ящика, из которого модуль почтового обмена будет забирать конверты по протоколу POP3. Адрес задается в соответствии со следующим правилом:

```
inputmailbox= <username>:<password>@<IP-адрес POP3-сервера>
```

- `outputmailbox` – IP-адрес SMTP-сервера, на который модуль почтового обмена будет отправлять конверты по протоколу SMTP.
- `frommailbox` – почтовый адрес отправителя SMTP-конвертов. Адрес задается в соответствии со следующим правилом:

```
frommailbox= <username>@<servername>.<domain>
```
- `mail_in_path` – полный путь к каталогу, в который модуль почтового обмена помещает принятые конверты. По умолчанию значение параметра `basedir/smtpin`.
- `mail_in_chunks_path` – полный путь к каталогу, в который модуль почтового обмена помещает принятые фрагменты SMTP-конвертов в случае использования протокола 2.0 (см. ниже). По умолчанию значение параметра `basedir/smtpin/chunks`.
- `mail_out_path` – полный путь к каталогу, в котором транспортный модуль формирует заголовочные файлы на отправляемые конверты. По умолчанию значение параметра `basedir/smtpout`.
- `mail_call_timeout` – интервал (в секундах) периодического вызова модуля почтового обмена, то есть период опроса почтового ящика входящих конвертов и отправки исходящих конвертов по каналу SMTP. При значении параметра `-1` периодический вызов не производится. Однако при наличии в очереди исходящих конвертов, предназначенных для отправки по каналу SMTP, вызов будет производиться, если это не запрещено параметром `call_flag` соответствующего канала. По умолчанию значение параметра `-1`.
- `version` – версия протокола инкапсуляции конверта MFTR в почтовый конверт RFC-822, передаваемый по каналу SMTP. Возможные значения параметра: `1.0` и `2.0`. В случае использования протокола `1.0` MFTR-конверт полностью инкапсулируется в SMTP-конверт для передачи. Версия протокола `2.0` позволяет передать MFTR-конверт в виде нескольких SMTP-конвертов. Это удобно в случае использования ограничений на размер писем SMTP/POP3-серверами. При отправке MFTR-конверт разбивается на несколько SMTP-конвертов, размер каждого из которых не превышает заданный параметром `maxsize` (см. ниже). Принимающая сторона собирает все SMTP-фрагменты в единый MFTR-конверт.

Параметр `version` может быть указан как в секциях для SMTP-каналов (см. «[Секция \[channel\]](#)» на стр. 151), так и глобально в секции `[mailtrans]`. По умолчанию данный параметр отсутствует в секциях `[channel]`. В случае отсутствия параметра в секции `[channel]` используется значение глобального параметра.



Примечание. По умолчанию используется протокол 1.0, так как предыдущие версии транспортного модуля несовместимы с протоколом 2.0.

- `maxsize` – максимальный размер почтового SMTP-конверта при отправке (в килобайтах). Используется в случае, если установлена версия протокола 2.0 (см. параметр `version`). Значение 0 означает, что ограничение на размер SMTP-конвертов отсутствует. Допустимые значения данного параметра: от 100 до 2048000 (2 ГБ). По умолчанию значение параметра 0.

Параметр `maxsize` может быть указан как в секциях для SMTP-каналов (см. «Секция [\[channel\]](#)» на стр. 151), так и глобально в секции `[mailtrans]`. По умолчанию данный параметр отсутствует в секциях `[channel]`. В случае отсутствия параметра в секции `[channel]` используется значение глобального параметра.

Секция `[journal]`

Секция `[journal]` содержит параметры настройки журнала MFTP-конвертов, обрабатываемых транспортным модулем. В процессе работы транспортный модуль осуществляет запись информации об обработанных конвертах в специальную базу данных, называемую журналом конвертов. В журнал заносится информация:

- о полностью принятых конвертах;
- об отправленных конвертах;
- об удаленных конвертах;
- о поврежденных конвертах.

Просмотр журнала конвертов осуществляется с помощью команды `mftp view` (см. «Команды группы `mftp`» на стр. 79).

Секция `[journal]` содержит следующие параметры:

- `use_journal` – включение/выключение ведения журнала в текущем сеансе работы транспортного модуля (`yes/no`). По умолчанию значение параметра `yes`. Если данный параметр отсутствует, то используется значение по умолчанию.
- `max_size` – максимальный размер (в мегабайтах) файла журнала конвертов. При превышении указанного размера осуществляется запись поверх самых старых записей. При изменении максимального размера журнала в процессе работы происходит реконструкция файла. В случае уменьшения размера по сравнению с

предыдущим из файла удаляются записи с наиболее старыми датами. По умолчанию значение параметра 1 (мегабайт). Если данный параметр отсутствует, то используется значение по умолчанию.

- `dump_filename` – префиксная часть имени текстового файла, в который осуществляется регулярный дамп информации из журнала конвертов. По умолчанию значение параметра `/var/log/mftpenv.log`. Постфиксная часть имени файла определяется текущей датой и зависит от интервала дампа, заданного параметром `dump_interval`. Например, файл дампа может иметь имя `/var/log/mftpenv.log.2009.09.23`.



Внимание! Менять параметр `dump_filename` вручную не следует!

- `dump_interval` – интервал создания (в днях) файлов дампа информации из журнала конвертов. В процессе работы транспортный модуль выводит информацию об обработанных конвертах в текущий файл дампа. По истечении интервала, заданного данным параметром, создается новый файл дампа, постфиксная часть которого определяется текущей датой. По умолчанию значение параметра 1 (день). Если данный параметр отсутствует, то используется значение по умолчанию.
- `last_dump` – время создания текущего файла дампа.



Внимание! Менять параметр `last_dump` вручную не следует!

Секция [misc]

Секция [misc] содержит различные параметры, определяющие работу транспортного модуля в целом:

- `port` – порт, на котором демон `mftpd` ожидает соединения по каналу MFTR от удаленных сетевых узлов. По умолчанию значение параметра 5000.
- `max_listen_ports` – диапазон значений перебора портов для соединений по каналу MFTR с удаленным узлом в случае неудачи. Транспортный модуль циклично перебирает порты в диапазоне от `port` до `port+max_listen_ports-1`. Для ожидания входящих соединений транспортный модуль прослушивает все порты из указанного диапазона. По умолчанию значение параметра 3.

- `num_attempts` – количество последовательных попыток соединения, после которых устанавливается тайм-аут, если соединиться так и не удалось. По умолчанию значение параметра 3.
- `max_connections` – максимальное количество входящих и исходящих соединений по каналам MFTR. По умолчанию значение параметра 100.
- `send_buff_size` – размер буфера передачи (в байтах). Минимально допустимое значение 1024 (байт), значение по умолчанию 65500 (байт).
- `recv_buff_size` – размер буфера приема (в байтах). Минимально допустимое значение 1024 (байт), значение по умолчанию 65500 (байт).

Во многих случаях значение 65500 параметров `send_buff_size` и `recv_buff_size` является оптимальным для обеспечения максимальной скорости приема/передачи конвертов транспортным модулем.

- `pingpong` – включение/выключение режима обмена конвертами по каналу MFTR (`yes/no`). По умолчанию значение параметра `yes`.

Если значение параметра `pingpong` установлено в `yes`, это означает, что сторона, передавшая конверт, позволяет передать конверт другой стороне, то есть узлы обмениваются конвертами поочередно. Если же значение установлено в `no`, то сторона, начавшая передавать конверты, будет их передавать, пока они не закончатся, и только после этого позволит передавать конверты другой стороне.

- `connect_timeout` – интервал (в секундах), в течение которого Координатор будет пытаться установить соединение с удаленным узлом по каналу MFTR. Если по истечении этого интервала соединение не было установлено, то повторные попытки будут происходить через интервал `outenv_timeout` (см. ниже). По умолчанию значение параметра 2 (секунды).
- `wait_timeout` – интервал (в секундах) ожидания активности установленного MFTR-соединения. Если в течение этого интервала узлы, установившие соединение, не обменялись никакой информацией, то данное соединение закрывается. Если в процессе обмена исходящие конверты для удаленного узла были переданы не полностью, то повторные попытки соединения будут происходить через интервал `outenv_timeout` (см. ниже). По умолчанию значение параметра 300 (секунд).
- `outenv_timeout` – интервал (в секундах), в течение которого исходящие конверты для канала, на котором произошла ошибка передачи, не могут быть повторно отправлены. Если на каком-либо канале произошла ошибка передачи (разрыв соединения и т.п.) и для этого канала существуют исходящие конверты, то следующая попытка передачи произойдет через `outenv_timeout` секунд. По умолчанию значение параметра 300 (секунд).

- `ttl_out` – время жизни конвертов в исходящей очереди (в днях). Если по истечении времени `ttl_out` конверт не удалось отправить, то он удаляется из очереди и помещается в корзину. По умолчанию значение параметра 30 (дней).
- `ttl_trash` – максимальное время хранения конвертов в корзине (в днях). Если время хранения конверта в корзине превышает `ttl_trash`, то конверт удаляется. По умолчанию значение параметра 90 (дней).
- `save_sent` – включение/выключение хранения имен отправленных прикладных конвертов (`yes/no`). По умолчанию значение параметра `no`. Если значение параметра установлено в `yes`, то при успешной отправке конверта в каталоге `out_path/sent` создается файл нулевой длины с именем, идентичным имени отправленного конверта.

Секция [debug]

Секция [debug] определяет параметры ведения журнала устранения неполадок транспортного модуля (см. «[Журналы устранения неполадок ПО ViPNet](#)» на стр. 213). Она содержит следующие параметры:

- `debuglevel` – уровень протоколирования, число от -1 до 5. Для модификаций ПАК с жестким диском значение по умолчанию 3. Для ПАК ViPNet Coordinator HW100 базовой конфигурации (без жесткого диска) значение по умолчанию -1. Значение параметра -1 отключает ведение журнала.
- `debuglogfile` – место хранения журнала, заданное в виде `syslog:<facility.level>`. По умолчанию значение параметра устанавливается в `syslog:daemon.debug`.



8

Настройка системного времени

О настройке системного времени	162
Списки континентов, стран и временных зон	163

О настройке системного времени

Для корректной работы ПАК ViPNet Coordinator HW в составе географически распределенной сети ViPNet требуется установить на нем точное системное время. Для этого необходимо сначала задать временную зону (часовой пояс) географической точки, в которой эксплуатируется ПАК, а затем установить текущее время. Эти установки производятся в процессе первоначального развертывания ключевых баз (см. «Первоначальное развертывание ключей» на стр. 42), но в дальнейшем могут быть изменены с помощью команд.

Установка временной зоны осуществляется с помощью команды `machine set timezone`, установка текущего времени – с помощью команды `machine set date` (см. «Команды группы machine» на стр. 82). Перед установкой времени требуется остановить все демоны, а после установки запустить их вновь с помощью соответствующих команд. Для просмотра текущих установок используются команды `machine show timezone` и `machine show date`.

Системное время можно периодически синхронизировать с NTP-серверами точного времени. По умолчанию синхронизация времени на ПАК включена, в качестве серверов точного времени используются публичные NTP-серверы из кластера `pool.ntp.org`. В процессе первоначального развертывания ключевых баз можно задать IP-адрес дополнительного NTP-сервера (например, корпоративного). В дальнейшем включение и выключение синхронизации, а также управление списком используемых NTP-серверов осуществляется с помощью команд (см. «Команды подгруппы inet ntp» на стр. 75). Подробнее о синхронизации времени см. раздел [Использование ПАК в качестве NTP-сервера](#) (на стр. 193).

Списки континентов, стран и временных зон

Установка временной зоны с помощью команды `machine set timezone` происходит в три этапа: сначала предлагается выбрать нужный континент, затем страну и далее собственно временную зону. Список континентов неизменный, он включает в себя также океаны. Список стран и список временных зон изменяются в зависимости от выбора, сделанного на предыдущем этапе. Все списки содержат английские названия согласно стандарту ISO 3166 и упорядочены по алфавиту.

Таблица ниже содержит полный список континентов и океанов, в двух следующих таблицах приведены примеры списков стран и временных зон.

Таблица 7. Список континентов и океанов

Порядковый номер	Английское название	Русское название
1	Africa	Африка
2	Americas	Америка (Северная и Южная)
3	Antarctica	Антарктика
4	Arctic Ocean	Ледовитый океан
5	Asia	Азия
6	Atlantic Ocean	Атлантический океан
7	Australia	Австралия
8	Europe	Европа
9	Indian Ocean	Индийский океан
10	Pacific Ocean	Тихий океан

Таблица 8. Список первых 10-ти стран Европы

Порядковый номер	Английское название	Русское название
1	Aaland Islands	Аландские острова
2	Albania	Албания
3	Andorra	Андорра
4	Austria	Австрия

5	Belarus	Беларусь
6	Belgium	Бельгия
7	Bosnia and Herzegovina	Босния и Герцеговина
8	Britain (UK)	Британия (Объединенное Королевство)
9	Bulgaria	Болгария
10	Croatia	Хорватия

Таблица 9. Список первых 5-ти временных зон в России

Порядковый номер	Английское название	Русское название
1	Moscow-01 – Kaliningrad	Калининград
2	Moscow+00 – west Russia	Западная Россия
3	Moscow+00 – Caspian Sea	Каспийское море
4	Moscow – Samara, Udmurtia	Самара
5	Moscow+02 – Urals	Урал



9

Удаленный мониторинг и управление ПАК

Мониторинг и управление ПАК с помощью апплета SGA	166
Настройка доступа к удаленному мониторингу и управлению	167

Мониторинг и управление ПАК с помощью апплета SGA

В состав ПО ПАК ViPNet Coordinator HW входит веб-сервер с апплетом SGA (Security Gateway Applet), который обеспечивает мониторинг и управление ПАК посредством веб-интерфейса. Мониторинг и управление ПАК с помощью апплета можно осуществлять как локально на самом ПАК, так и удаленно на других узлах сети ViPNet.

Локальный запуск апплета возможен сразу после установки ПО на ПАК, без каких-либо дополнительных настроек.

Удаленный запуск апплета возможен на узлах, удовлетворяющих определенным требованиям. При этом доступ к функциям мониторинга и управления на таких узлах зависит от настроек, заданных на самом ПАК (см. [«Настройка доступа к удаленному мониторингу и управлению»](#) на стр. 167).

Описание работы с апплетом SGA содержится в документе «Апплет мониторинга и управления ViPNet-координатором. Руководство пользователя».

Настройка доступа к удаленному мониторингу и управлению

Удаленный мониторинг и управление ПАК ViPNet Coordinator HW с помощью апплета SGA может осуществляться на сетевых узлах, удовлетворяющих следующим требованиям:

- Узлы являются Клиентами ViPNet (с установленным ПО ViPNet Client [Монитор]).
- Клиенты ViPNet зарегистрированы в прикладной задаче «Клиент SGA» и имеют связь с ПАК ViPNet Coordinator HW.

Регистрация узлов в прикладных задачах и задание связей между узлами осуществляются в ЦУСе.

По умолчанию предполагается, что всем сетевым узлам, зарегистрированным в прикладной задаче «Клиент SGA» и связанным с ПАК, разрешены мониторинг и управление ПАК. Однако возможны ситуации, когда требуется запретить возможность запуска апплета на определенных узлах.



Примечание. Сетевой узел, зарегистрированный в прикладной задаче «Клиент SGA», может быть связан с несколькими ViPNet-координаторами, на которых установлен апплет. При этом узел предполагается использовать для мониторинга и управления только одним определенным ViPNet-координатором, а связь с остальными ViPNet-координаторами требуется для других задач. В этом случае необходимо явно запретить этому узлу мониторинг и управление всеми ViPNet-координаторами, кроме одного.

Для ограничения доступа узлов к функциям мониторинга и управления ПАК с помощью апплета служит файл конфигурации `sga.conf`. Для редактирования этого файла используется команда `iplir config sga` (см. «Команды группы `iplir`» на стр. 76). Перед редактированием файла необходимо остановить все демоны, произвести необходимые изменения и затем снова запустить демоны с помощью соответствующих команд.

Файл содержит одну секцию `[access]`, в которой могут присутствовать следующие параметры:

- `allow` – список идентификаторов узлов (через запятую), которым разрешены мониторинг и управление ПАК.
- `deny` – список идентификаторов узлов (через запятую), которым запрещены мониторинг и управление ПАК.
- `default` – значение доступа к функциям мониторинга и управления по умолчанию. Этот параметр определяет доступ для всех узлов, которые не описаны явно параметрами `allow` или `deny`. Параметр `default` может принимать следующие значения:
 - `allow` – разрешить мониторинг и управление;
 - `deny` – запретить мониторинг и управление.

Значения параметров отделяются от их идентификаторов знаком равенства (=).

Параметры `allow` и `deny` могут встречаться в секции несколько раз, при этом их наличие не является обязательным. Идентификаторы узлов в этих параметрах могут повторяться. Параметр `default` должен присутствовать в секции обязательно и только один раз.

По умолчанию мониторинг и управление ПАК разрешены всем узлам: секция `[access]` содержит один параметр `default` со значением `allow`, параметры `allow` и `deny` отсутствуют.

Проверка доступа конкретного узла к функциям мониторинга и управления ПАК выполняется следующим образом:

- Если узел не зарегистрирован в прикладной задаче «Клиент SGA», то доступ запрещен.
- Если узел зарегистрирован в прикладной задаче «Клиент SGA», то анализируется файл `sga.conf`:
 - если идентификатор узла встречается в параметрах `allow` или `deny`, то доступ соответственно разрешен или запрещен;
 - если идентификатор узла встречается в нескольких параметрах `allow` или `deny`, то приоритет имеет тот параметр, которой находится в секции `[access]` ниже всех;
 - если идентификатор узла не встречается ни в одном из параметров `allow` и `deny`, то доступ разрешается или запрещается в зависимости от значения параметра `default`.



10

Система защиты от сбоев

Назначение системы защиты от сбоев	170
Состав системы защиты от сбоев и принципы ее работы	171
Управление системой защиты от сбоев	173
Настройка системы защиты от сбоев	174

Назначение системы защиты от сбоев

Система защиты от сбоев предназначена для создания отказоустойчивого решения на базе ПАК ViPNet Coordinator HW. Данная система имеет два режима функционирования:

1. Одиночный режим (режим одиночного ПАК).
2. Режим кластера (режим кластера горячего резервирования ПАК).



Внимание! Кластер горячего резервирования можно организовать только на базе ПАК ViPNet Coordinator HW1000 и HW-VPNМ. ПАК ViPNet Coordinator HW100 можно использовать только как одиночный ПАК, поэтому на нем система защиты от сбоев всегда функционирует в одиночном режиме.

При работе в одиночном режиме, который устанавливается автоматически при установке ПО ПАК ViPNet Coordinator HW, система защиты от сбоев выполняет функции, обеспечивающие постоянную работоспособность основных служб, входящих в состав ПО:

- постоянный контроль состояния служб и ведение статистики использования системных ресурсов;
- обнаружение факта сбоя службы и осуществление последующих попыток восстановления работоспособности сбойного приложения;
- предотвращение внутренних сбоев в работе самой системы защиты от сбоев;
- предотвращение сбоев при обработке пакетов драйвером сетевой защиты iplic.

Режим кластера горячего резервирования обеспечивает передачу функций вышедшего из строя ПАК другому (резервному) ПАК. При работе в режиме кластера система защиты от сбоев также выполняет функции одиночного режима, т.е. обеспечивает постоянную работоспособность основных служб, входящих в состав ПО. Подробное описание режима кластера приведено в документе «ПАК ViPNet Coordinator HW. Система защиты от сбоев. Руководство администратора». В данном документе содержится описание одиночного режима работы системы защиты от сбоев.

Состав системы защиты от сбоев и принципы ее работы

Система защиты от сбоев состоит из драйвера и программы-демона, работающей в фоновом режиме. Драйвер `watchdog` работает на низком уровне и в большинстве случаев сохраняет работоспособность даже в случаях, когда система уже не реагирует на внешние события. В зависимости от настройки параметра `reboot` (см. «[Настройка системы защиты от сбоев](#)» на стр. 174) программа-демон `failoverd` при запуске регистрируется в драйвере и периодически опрашивает его, подтверждая работоспособность системы. Если по истечении заданного промежутка времени драйвер обнаруживает, что опроса не было, то он перезагружает систему. Перед этим он делает попытку записать на диск кэш-буферы системы, чтобы не возникло ошибок в файловой системе, однако это не всегда возможно. При корректной остановке программы-демона (например, для изменения настроек системы защиты от сбоев) она сообщает драйверу об этом, и драйвер перестает следить за временем опроса, так что система не будет перезагружена. Такой механизм обеспечивает предотвращение внутренних сбоев в демоне `failoverd`.

При старте ОС демон системы защиты от сбоев `failoverd` осуществляет старт подконтрольных служб, а также дальнейшее слежение за ними. Демон `failoverd` постоянно контролирует работоспособность следующих служб ViPNet:

- управляющий демон (`iplircfg`);
- транспортный модуль MFTP (`mftpd`).

Контроль работы этих служб осуществляется путем их регистрации в системе защиты от сбоев в момент старта с установкой периода оповещения. В процессе работы контролируемая служба (приложение) периодически определяет свое состояние и оповещает о нем систему слежения. Если контролируемое приложение в течение периода оповещения не сообщило о своем состоянии или сообщило о внутреннем сбое, то система защиты от сбоев идентифицирует сбой приложения и инициирует процедуру восстановления работоспособности этого приложения. Для этого сначала делается попытка корректной остановки сбойного приложения. Если эта попытка оказывается неудачной, то осуществляется принудительная «некорректная» остановка приложения. После этого система защита от сбоев перезапускает остановленное приложение.

В процессе работы демон `failoverd` ведет статистику сбоев для каждого контролируемого приложения, в том числе и для самого себя. Если обнаруживается, что для какого-либо из приложений произошло 5 сбоев подряд, т.е. в течении 5-и попыток восстановления

работоспособности приложение не смогло корректно стартовать, то делается вывод о полной неработоспособности приложения. В этом случае, в зависимости от настроек системы защиты от сбоев (см. «[Настройка системы защиты от сбоев](#)» на стр. 174), производится либо перезагрузка ОС, либо остановка сбойного приложения и прекращение слежения за ним.

Система защиты от сбоев отслеживает также сбои, которые могут произойти в потоках обработки пакетов драйвера сетевой защиты `iplir`. Для этого как следящее, так и контролируемые приложения при старте осуществляют специальный запрос к драйверу `iplir`. Если в ответ на этот запрос был получен код ошибки, соответствующий сбою одного из потоков обработки пакетов в драйвере, то контролируемое приложение сообщает факт внутреннего сбоя следящему приложению, которое в свою очередь обрабатывает стандартную логику, описанную выше. Помимо старта, управляющий демон осуществляет периодические запросы к драйверу `iplir` и в процессе своей работы (запрос информации о журнале пакетов). При этом логика обнаружения сбоев в потоках обработки пакетов такая же, как при старте контролируемого приложения. Описанный механизм позволяет оперативно (от нескольких десятков секунд до нескольких минут – в зависимости от производительности компьютера и ряда других внешних факторов) отследить факт сбоя в работе драйвера и осуществить корректирующие действия (перезагрузить компьютер в случае включения соответствующей настройки). Однако этот механизм не сможет отследить все возможные сбои в драйвере `iplir`, например, зависание одного из потоков обработки и т.д. Поэтому его нельзя расценивать как универсальное средство от любого типа сбоев уровня драйверов.

Если контролируемое приложение было корректно остановлено администратором системы с помощью соответствующей команды (`iplir stop` или `mftp stop`), то оно производит deregистрацию в системе защиты от сбоев, слежение за ним отключается. В этом случае для дальнейшей работы администратор должен вручную запустить приложение (соответственно командой `iplir start` или `mftp start`).

Если при запуске демона `failoverd` выясняется, что какие-либо из подконтрольных демонов были остановлены вручную, то об этом выдается предупреждение в `syslog`, а также на терминал, если он есть. Предупреждение выдается также в случае, если в течение 10-и проверок подряд одного демона он находится в режиме ручной остановки.

Управление системой защиты от сбоев

Набор действий администратора по управлению системой защиты от сбоев в одиночном режиме работы сведен к минимуму: администратор может запустить или остановить демон `failoverd`. Управление производится с помощью команд из группы `failover` (см. «Команды группы `failover`» на стр. 78).

Запуск системы защиты от сбоев производится командой `failover start`. При этом запускается демон `failoverd`. Остановка системы производится командой `failover stop`. При этом демон `failoverd` завершает работу, сообщая об этом драйверу.

При загрузке системы автоматически загружается драйвер `watchdog` и выполняется команда `failover start`. В результате загружается демон `failoverd`, который в свою очередь производит старт остальных контролируемых служб ViPNet.



Внимание! При выполнении команды `failover start` вручную в одиночном режиме перезапуск контролируемых приложений не производится.

Для запроса информации о работе системы защиты от сбоев используется команда `failover show info`.

Настройка системы защиты от сбоев

Администратор может настраивать параметры работы системы защиты от сбоев путем редактирования файла конфигурации. Для редактирования файла конфигурации системы защиты от сбоев используется команда `failover config edit` (см. «Команды группы [failover](#)» на стр. 78).

Файл конфигурации системы защиты от сбоев состоит из нескольких секций. Каждая секция начинается со строки, содержащей имя секции в квадратных скобках. Каждая секция содержит несколько параметров. Строка с параметром начинается с имени параметра, затем идет знак «`=`» и пробел, затем значение этого параметра. Для настройки системы в одиночном режиме работы служат следующие параметры:

- Секция `[misc]` содержит параметр, отвечающий за действия системы при обнаружении полной неработоспособности любого из контролируемых приложений:
 - `reboot` – указывает, должен ли демон `failoverd` включать механизм регистрации в драйвере `watchdog` и должна ли производиться перезагрузка ОС в случае, если какое-либо из контролируемых приложений не может восстановить свою работоспособность (см. «Состав системы защиты от сбоев и принципы ее работы» на стр. 171). Может принимать значение `yes` или `no`. Значение `yes` включает механизм перезагрузки системы, `no` – выключает. Параметр является обязательным.
- Секция `[debug]` определяет параметры ведения журнала устранения неполадок демона `failoverd` (см. «Журналы устранения неполадок ПО ViPNet» на стр. 213). Она содержит следующие параметры:
 - `debuglevel` – уровень протоколирования, число от -1 до 5. Для модификаций ПАК с жестким диском значение по умолчанию 3. Для ПАК ViPNet Coordinator HW100 базовой конфигурации (без жесткого диска) значение по умолчанию -1. Значение параметра -1 отключает ведение журнала.
 - `debuglogfile` – место хранения журнала, заданное в виде `syslog:<facility.level>`. По умолчанию значение параметра устанавливается в `syslog:daemon.debug`.

Перед редактированием файла конфигурации нужно остановить демон `failoverd` командой `failover stop`, а после редактирования запустить его снова командой `failover start`.

Для просмотра конфигурации системы защиты от сбоев используется команда `failover show config` (см. «Команды группы [failover](#)» на стр. 78).



11

Работа с конфигурациями ViPNet

О конфигурациях ViPNet	176
Команды для работы с конфигурациями ViPNet	177

О конфигурациях ViPNet

Конфигурация ViPNet – это совокупность файлов конфигурации, справочников и ключевых баз. Различают два типа конфигурации:

- **Частичная конфигурация (partial).**

Частичная конфигурация включает в себя все конфигурационные файлы компонентов ViPNet.

- **Полная конфигурация (full).**

Полная конфигурация включает в себя все файлы частичной конфигурации, а также все справочники и ключевые базы ViPNet.

Для защиты файлов конфигурации, справочников и ключевых баз от искажения в состав ПО ViPNet включена система сохранения конфигураций, которая позволяет выполнять следующее:

- Сохранить текущую конфигурацию.

При соответствующих настройках транспортного модуля (см. «Секция [upgrade]» на стр. 154) предусмотрено автоматическое сохранение текущей конфигурации перед проведением обновления – такая конфигурация называется автосохраненной конфигурацией.

- Загрузить одну из ранее сохраненных пользователем конфигураций.

Загрузка сохраненной конфигурации осуществляется только вручную по команде пользователя, при этом текущая конфигурация теряется. Однако перед загрузкой пользователю предлагается сохранить текущую конфигурацию для возможного использования в будущем.

- Удалить сохраненную пользователем конфигурацию.

Удаление сохраненных конфигураций производится только вручную по команде пользователя.

Команды для работы с конфигурациями ViPNet

Для работы с конфигурациями ViPNet предназначены следующие команды:

- `admin config list [<версия>]` – просмотр текущего набора сохраненных конфигураций в следующем формате:

«Уникальное имя конфигурации», версия ПО ViPNet, с помощью которой была создана данная конфигурация (может отсутствовать), тип конфигурации, дата и время сохранения, дата и время загрузки.

Например:

```
"Config_1", full, saved on 21.03.2011 at 11:49, never loaded
"autosave-2011-1-0", full, saved on 31.03.2011 at 17:05, loaded at
01.04.2011 at 12:45
"rollback-2011-10-12", version 1.0.3, part, saved on 12.10.2011 at
11:30, never loaded
```

`версия` – необязательный параметр, позволяющий фильтровать запрашиваемые конфигурации по версии ПО ViPNet, с помощью которой была создана данная конфигурация. Задается в следующем формате: `Major.Minor.Subminor`. Если данный параметр присутствует, то выводится информация только о тех конфигурациях, у которых версия ПО меньше или равна заданному параметру.

В случае автосохранения имя конфигурации начинается со слова `autosave`.

- `admin config save <имя>` – сохранение текущей конфигурации.

`имя` – имя, под которым нужно сохранить конфигурацию.

Если уже существует сохраненная конфигурация с заданным именем, то командный интерпретатор запрашивает пользователя, нужно ли перезаписать имеющуюся конфигурацию.



Внимание! В именах конфигураций можно использовать только символы латинского алфавита, цифры, знаки дефис и подчеркивание.

- `admin config delete <имя> [<версия>]` – удаление сохраненной конфигурации с заданным именем `имя`. В имени можно указать символ подстановки «*», который

обозначает любое количество символов. Таким образом, можно производить удаление конфигураций по шаблону имени. Например:

```
admin config delete "Config_*"
admin config delete "autosave-2011-08-03-*" 1.0.3
```

Если под заданный шаблон имени подходит более одной конфигурации, то выдается предупреждение и у пользователя запрашивается подтверждение на удаление.



Внимание! При использовании символа подстановки необходимо заключить параметр в двойные кавычки. В противном случае команда может быть неправильно интерпретирована, что приведет к некорректным результатам работы.

версия – необязательный параметр, позволяющий указать версию ПО, к которой относится конфигурация. Задается в следующем формате: Major.Minor.Subminor. Если параметр не указан и существует несколько конфигураций с совпадающими именами, но различными версиями, то команда не выполняется, пользователю выводится список имеющихся конфигураций с предложением указать версию.

- `admin config load <имя> [<версия>]` – восстановление конфигурации с заданным именем *имя* и версией *версия*. Перед восстановлением конфигурации командный интерпретатор спрашивает пользователя, хочет ли он сохранить текущую конфигурацию. Рекомендуется согласиться и ввести имя, под которым будет сохранена текущая конфигурация до восстановления. Если пользователь отказывается сохранять текущую конфигурацию, то программа во избежание ошибок требует ввести специальную фразу. Например:

```
Save current configuration [y/n]? n
You are about to overwrite your existing configuration.
This is unsafe. To continue, type
>>> Yes, do as I say
in response to the prompt:
>>>
```

В этом случае пользователю необходимо ввести фразу «Yes, do as I say» для продолжения восстановления. В противном случае надо нажать клавишу **<Enter>**, чтобы отменить восстановление.

версия – необязательный параметр, позволяющий указать версию ПО, к которой относится конфигурация. Задается в следующем формате: Major.Minor.Subminor. Если параметр не указан и существует несколько конфигураций с совпадающими именами, но различными версиями, то команда не выполняется, пользователю выводится список имеющихся конфигураций с предложением указать версию.

При попытке восстановления конфигурации текущая версия ПО ViPNet сравнивается с версией, к которой относится конфигурация. Если текущая версия ПО больше или равна версии, к которой относится конфигурация, то производится восстановление без дополнительного подтверждения. В противном случае пользователю сообщается, что конфигурация относится к более поздней версии ПО, и, возможно, потребуются вручную отредактировать файлы конфигурации, чтобы они были корректно восприняты. Затем пользователю задается вопрос, действительно ли он хочет восстановить эту конфигурацию. При отрицательном ответе конфигурация не восстанавливается.

Перед проведением операций сохранения и восстановления конфигураций должны быть остановлены все службы ViPNet. Если это условие не выполняется, то появится соответствующее сообщение и команда не будет выполняться.



12

Экспорт и импорт ключевых баз, справочников и настроек

Экспорт и импорт ключевых баз, справочников и настроек	181
Выполнение экспорта ключевых баз, справочников и настроек	183

Экспорт и импорт ключевых баз, справочников и настроек

Процедуры экспорта и импорта ключевых баз, справочников и настроек служб ViPNet предназначены соответственно для сохранения и восстановления этих данных на ПАК в случае обновления на нем версии ПО ViPNet, а также для переноса ключевых баз, справочников и настроек с одного действующего ПАК на другой. Применение этих процедур позволяет упростить подготовку ПАК к работе, так как при их использовании не требуется выполнять ручную настройку служб ViPNet, параметров интерфейсов и маршрутов.

Экспорт ключевых баз, справочников и настроек осуществляется в файл, из которого их можно импортировать на другой действующий ПАК, имеющий такой же тип лицензии. Импорт ключевых баз, справочников и настроек выполняется в рамках процедуры первоначального развертывания ключей (см. «[Первоначальное развертывание ключей](#)» на стр. 42).



Внимание! Во время выполнения экспорта ПАК окажется незащищенным, так как до начала экспорта требуется вручную остановить все демоны.

В состав экспортируемой информации входят следующие данные:

- файлы, содержащие ключи шифрования для связанных с ПАК узлов (ключевые базы);
- файлы, содержащие информацию о связанных с ПАК узлах (справочники);
- настройки защищенной сети (файл `iplir.conf`);
- настройки сетевых интерфейсов (IP-адрес, маска подсети, файл `iplir.conf`-`<интерфейс>`);
- маршрут по умолчанию и статические маршруты (системная таблица маршрутизации);
- настройки временной зоны;
- правила обработки открытых IP-пакетов (файл `firewall.conf`);

- настройки системы защиты от сбоев;
- настройки транспортного модуля (файл `mftp.conf`);
- настройки протоколирования;
- настройки дополнительных сервисов (DHCP-сервера, DNS-сервера, NTP-сервера);
- настройки взаимодействия с UPS;
- текущие ключи для соединений по протоколу SSH2;
- журнал регистрации IP-пакетов;
- журнал транспортных конвертов MFTP;
- очередь почтовых конвертов (для ПАК, поддерживающих функцию Сервера-маршрутизатора).

Экспорт и импорт осуществляются с использованием ноутбука, подключенного к порту Ethernet ПАК с помощью кроссированного кабеля Ethernet, или с использованием флэш-памяти USB, чтобы исключить передачу ключей через незащищенные каналы. Файл экспорта шифруется на пароле пользователя, который необходимо будет ввести при последующем выполнении импорта.

Для восстановления ключевых баз, справочников и настроек после обновления на ПАК версии ПО ViPNet необходимо выполнить процедуру первоначального развертывания ключей, выбрав режим импорта и указав файл экспорта в качестве источника импорта (см. «[Первоначальное развертывание ключей](#)» на стр. 42). Эту же процедуру необходимо выполнить для переноса (импорта) на действующий ПАК ключевых баз, справочников и настроек, полученных в результате экспорта на другом ПАК. В последнем случае перед выполнением импорта необходимо удалить текущие ключи, справочники и настройки с помощью команды `admin remove keys` (см. «[Команды группы admin](#)» на стр. 79).

Выполнение экспорта ключевых баз, справочников и настроек

Экспорт ключевых баз, справочников и настроек можно выполнить одним из следующих способов:

- С помощью мобильного компьютера (ноутбука), на котором установлена сетевая карта Ethernet и ОС Windows XP или Windows Vista.
- С помощью флэш-памяти USB (USB-флэш), которая отформатирована в одну из поддерживаемых файловых систем: FAT32 или ext2.

При экспорте с помощью ноутбука используется стандартная служба TFTP. В ОС Windows XP эта служба по умолчанию включена. В ОС Windows Vista эта служба по умолчанию отключена и ее необходимо включить вручную. Для включения службы в ОС Windows Vista выполните следующее:

- 1 Выберите **Start > Control Panel > Programs and Features**.
- 2 Зайдите в меню **Turn Windows features on or off** и включите службы **TFTP Client** и **Simple TCP/IP services**.

Кроме того, на время выполнения экспорта на ноутбуке с ОС Windows Vista отключите следующие службы безопасности (если они включены):

- Windows Firewall;
- Windows Defender;
- Windows Update;
- отключите защиту по всем параметрам в Internet Explorer (меню **Internet Options**, закладка **Security**).



Примечание. Управление процедурой экспорта осуществляется только с одной из консолей. Удаленное управление экспортом по протоколу Telnet невозможно.

На время выполнения экспорта с помощью ноутбука автоматически изменяются настройки сетевых интерфейсов ПАК: на интерфейсе Ethernet1 устанавливается IP-адрес 169.254.241.1, все остальные интерфейсы ПАК выключаются. После успешного завершения экспорта настройки интерфейсов ПАК будут автоматически восстановлены.



Внимание! Во избежание потери удаленного доступа к ПАК запрещается выполнять экспорт на ноутбук по протоколу TFTP в удаленной SSH-сессии.

Процедура экспорта выполняется в следующей последовательности:

- 1 Подключите ноутбук к порту Ethernet1 ПАК с помощью кроссированного кабеля Ethernet или вставьте USB-флэш в USB-разъем ПАК.
- 2 При экспорте с помощью ноутбука установите вручную на сетевом интерфейсе ноутбука IP-адрес 169.254.241.5.
- 3 Подключите к ПАК СОМ-консоль или обычную консоль (монитор и клавиатуру).
- 4 Остановите все демоны с помощью соответствующих команд.
- 5 Выполните команду `admin export keys binary-encrypted` (см. «Команды группы admin» на стр. 79), указав в ней соответствующий параметр (`tftp` или `usb`).

При успешном выполнении команды появляется сообщение, содержащее имя сформированного файла экспорта, и предложение скачать этот файл.



Примечание. Имя файла экспорта формируется по следующему шаблону:

<модификация ПАК>-<идентификатор узла>-<дата экспорта>.vbe.

- 6 Перенесите файл экспорта на ноутбук или на USB-флэш. Перенос файла на ноутбук осуществляется по TFTP с помощью команды:

```
tftp -i 169.254.241.1 get <имя файла>
```

- 7 Нажмите ввод.



Внимание! Запись файла экспорта на USB-флэш завершается сообщением, разрешающим извлечение USB-флэш. Необходимо дождаться этого сообщения прежде, чем извлечь USB-флэш из разъема.

После завершения процедуры экспорта необходимо запустить все демоны с помощью соответствующих команд.

13

Контроль целостности конфигурационных файлов

Конфигурационные файлы ОС Linux и служб, входящих в состав ПАК ViPNet Coordinator HW, защищены контрольными суммами, которые хранятся в файлах с расширением *.crg. Каждой службе ViPNet сопоставлен один файл *.crg, содержащий контрольные суммы всех конфигурационных файлов, используемых этой службой. В отдельном файле *.crg хранятся контрольные суммы файлов конфигурации ОС Linux и пакета NUT. Каждый файл *.crg содержит также собственную контрольную сумму. Проверка контрольной суммы файла *.crg выполняется перед проверкой целостности файлов конфигурации.

Проверка целостности конфигурационных файлов выполняется при каждой попытке их использования:

- при запуске демонов и командного интерпретатора;
- при выполнении команд редактирования файлов конфигурации демонов;
- при выполнении команд изменения параметров, хранящихся в файлах конфигурации ОС Linux;
- при выполнении процедуры регламентного тестирования, инициируемой командой `machine self-test`.

Проверка целостности конфигурационного файла осуществляется путем вычисления текущей контрольной суммы файла и ее сравнения с контрольной суммой из файла *.crg. При несовпадении контрольных сумм выводится сообщение о нарушении целостности этого файла конфигурации. Нарушением целостности конфигурационных файлов является также отсутствие хотя бы одного файла конфигурации или файла с контрольными суммами. В любом случае нарушения целостности файлов конфигурации службы ViPNet запускаться не будут.



14

Сервисные функции

Использование ПАК в качестве DHCP-сервера	189
Использование ПАК в качестве DNS-сервера	191
Использование ПАК в качестве NTP-сервера	193
Взаимодействие ПАК с UPS	196

Использование ПАК в качестве DHCP-сервера

В состав ПО ПАК ViPNet Coordinator HW входит DHCP-сервер, который может использоваться для динамического назначения IP-адресов сетевым узлам (DHCP-клиентам). Одновременно с выделением IP-адресов DHCP-сервер может назначать дополнительные параметры настройки клиентов, например, IP-адреса шлюза по умолчанию и WINS-серверов. В качестве адресов DNS-сервера и NTP-сервера DHCP-сервер всегда предоставляет клиентам адрес интерфейса, с которым он работает. Клиенты, получившие от ПАК вместе с IP-адресом адреса DNS- и NTP-серверов, будут осуществлять запросы на разрешение имен и синхронизацию времени через соответствующие серверы, запущенные на ПАК. Если на ПАК эти серверы остановлены, то клиенты не смогут работать с DNS-именами и синхронизировать свое время.



Внимание! Использование ПАК в качестве DHCP-сервера возможно только при работе ПАК в одиночном режиме (см. «Система защиты от сбоев» на стр. 169). Работа DHCP-сервера в режиме кластера горячего резервирования не поддерживается.

При настройке DHCP-сервера должны соблюдаться следующие ограничения:

- DHCP-сервер может выделять IP-адреса только из диапазонов, установленных стандартом для частных сетей (допустимые диапазоны адресов): 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16.
- DHCP-сервер может использовать только интерфейс со статическим адресом, принадлежащим одному из допустимых диапазонов адресов.
- Назначенный DHCP-серверу интерфейс должен быть включен.
- Выделяемые клиентам IP-адреса должны входить в подсеть интерфейса, назначенного DHCP-серверу.
- Адрес интерфейса, назначенного DHCP-серверу, не должен входить в диапазон выделяемых клиентам адресов.

Настройка и управление DHCP-сервером осуществляется с помощью команд из группы `inet` (см. «Команды подгруппы `inet dhcp`» на стр. 73). Управление сводится к запуску и

остановке сервера. Запуск DHCP-сервера производится командой `inet dhcp start`, для остановки сервера используется команда `inet dhcp stop`. Кроме того, можно настроить автоматический запуск DHCP-сервера при старте ПАК с помощью команд `inet dhcp mode on` (включить автоматический запуск) и `inet dhcp mode off` (выключить автоматический запуск). Автоматический запуск DHCP-сервера по умолчанию выключен, и при первом после установки ПО старте ПАК DHCP-сервер не запускается.

Для настройки DHCP-сервера служат следующие параметры:

- Интерфейс, который должен использовать DHCP-сервер (команда `inet dhcp interface`).
- Диапазон адресов, доступных для назначения DHCP-клиентам (команда `inet dhcp range`).
- Время аренды (лизинга) IP-адресов, выделяемых DHCP-сервером клиентам (команда `inet dhcp lease`).

Время аренды означает срок, на который клиенту предоставляется IP-адрес. По истечении половины этого срока клиент должен возобновить аренду, обратившись к DHCP-серверу с повторным запросом. Таким образом, время аренды влияет на частоту обновления аренды, т.е. на интенсивность обращений к DHCP-серверу.

- IP-адрес шлюза по умолчанию (команда `inet dhcp router`).
- IP-адреса WINS-серверов. Можно задать адреса нескольких WINS-серверов, список адресов формируется с помощью команд добавления (`inet dhcp add wins`) и удаления (`inet dhcp delete wins`).

Изменение настроек DHCP-сервера запрещено, если сервер запущен. В этом случае все приведенные команды выполняться не будут.

В случае если DHCP-сервер остановлен, производимые командами изменения проверяются на соблюдение ограничений, приведенных выше. Если автоматический запуск DHCP-сервера включен, то при несоблюдении ограничений выдается сообщение об ошибке и изменения не применяются. Если автоматический запуск DHCP-сервера выключен, то при несоблюдении ограничений изменения применяются, при этом выдается сообщение о некорректности текущих настроек DHCP-сервера.

Для просмотра текущего состояния DHCP-сервера и его настроек используется команда `inet show dhcp` (см. «[Команды группы inet](#)» на стр. 67).

Использование ПАК в качестве DNS-сервера

В состав ПО ПАК ViPNet Coordinator HW входит DNS-сервер, который может использоваться для разрешения (преобразования) символьных имен в IP-адреса в ответ на собственные запросы и на запросы других сетевых узлов (DNS-клиентов).

По умолчанию DNS-сервер, установленный на ПАК (локальный DNS-сервер), настроен таким образом, что он может выполнять разрешение имен с использованием корневых DNS-серверов. Для этого требуется наличие подключения к Интернету. При отсутствии доступа к Интернету для разрешения имен следует использовать другой доступный (не корневой) DNS-сервер (серверы). В этом случае необходимо добавить адрес доступного сервера (серверов) в настройки локального DNS-сервера. Все DNS-серверы, отличные от корневых, добавляются в качестве форвардных (forwarder) серверов. Если адрес доступного DNS-сервера известен заранее, то его можно задать в процессе первоначального развертывания ключевых баз (см. «[Процедура развертывания ключевых баз](#)» на стр. 46).



Внимание! В ситуации, когда форвардные серверы, заданные в настройках локального DNS-сервера, недоступны, но при этом доступны корневые DNS-серверы, DNS-клиенты будут получать ответы на свои запросы с задержкой.

Список форвардных DNS-серверов можно сформировать вручную с помощью команд или получить их адреса от внешнего DHCP-сервера (если на одном из интерфейсов ПАК установлен режим DHCP). При этом полученный от DHCP-сервера список нельзя изменить с помощью команд. При получении адресов от DHCP-сервера полностью перезаписывается существующий список (независимо от способа его формирования). После перезагрузки ПАК или установки на интерфейсе статического IP-адреса список, полученный от внешнего DHCP-сервера, удаляется из настроек локального DNS-сервера. В этом случае для разрешения имен будут использоваться корневые DNS-серверы.



Примечание. Если режим DHCP установлен на нескольких интерфейсах ПАК, то результат обработки собственных DNS-запросов и запросов DNS-клиентов не определен, так как неизвестно, какой интерфейс будет включен первым и какой список форвардных DNS-серверов получит ПАК.

Настройка и управление DNS-сервером осуществляется с помощью команд из группы `inet` (см. «Команды подгруппы `inet dns`» на стр. 74). Управление сводится к запуску и остановке сервера. Запуск DNS-сервера производится командой `inet dns start`, для остановки сервера используется команда `inet dns stop`. Кроме того, можно настроить автоматический запуск DNS-сервера при старте ПАК с помощью команд `inet dns mode on` (включить автоматический запуск) и `inet dns mode off` (выключить автоматический запуск). Изменение настройки автоматического запуска не влияет на текущее состояние DNS-сервера, новая настройка вступает в силу только при следующем старте ПАК. Для изменения состояния DNS-сервера в текущем сеансе работы следует использовать команды `inet dns start` и `inet dns stop`. Автоматический запуск DNS-сервера по умолчанию включен, и при первом после установки ПО старте ПАК DNS-сервер запускается.

Формирование списка форвардных DNS-серверов осуществляется с помощью команд добавления (`inet dns add`) и удаления (`inet dns delete`). Для просмотра списка форвардных DNS-серверов используется команда `inet dns list`.

Если на ПАК запущен DHCP-сервер, то в качестве адреса DNS-сервера он всегда предоставляет DHCP-клиентам свой собственный адрес.

Для просмотра текущего состояния DNS-сервера и его настроек используется команда `inet show dns` (см. «Команды группы `inet`» на стр. 67).

Использование ПАК в качестве NTP-сервера

В состав ПО ПАК ViPNet Coordinator HW входит NTP-сервер, который может использоваться для синхронизации времени на самом ПАК и на других сетевых узлах (NTP-клиентах).

По умолчанию NTP-сервер, установленный на ПАК (локальный NTP-сервер), настроен таким образом, что при наличии подключения к Интернету он может осуществлять синхронизацию времени с использованием публичных NTP-серверов из кластера `pool.ntp.org`. Этот кластер серверов можно дополнить другими NTP-серверами (публичными или корпоративными). Такая необходимость может возникнуть в случае отсутствия доступа к Интернету или при наличии более близкого и менее нагруженного NTP-сервера (например, корпоративного). Если адрес дополнительного NTP-сервера известен заранее, то его можно задать в процессе первоначального развертывания ключевых баз (см. «[Процедура развертывания ключевых баз](#)» на стр. 46).



Примечание. Если в качестве дополнительного NTP-сервера используется защищенный узел, видимый по реальному адресу, то для успешной синхронизации времени с таким сервером при старте системы в файле `iplir.conf` на ПАК в секции `[id]` для этого защищенного узла рекомендуется установить параметр `forcereal` в значение `on`.

Список дополнительных NTP-серверов можно сформировать вручную с помощью команд или получить их адреса от внешнего DHCP-сервера (если на одном из интерфейсов ПАК установлен режим DHCP). При этом полученный от DHCP-сервера список нельзя изменить с помощью команд. При получении адресов от DHCP-сервера полностью перезаписывается существующий список (кроме заданного по умолчанию кластера серверов `pool.ntp.org`). После перезагрузки ПАК или установки на интерфейсе статического IP-адреса список, полученный от внешнего DHCP-сервера, удаляется из настроек локального NTP-сервера.



Примечание. Если режим DHCP установлен на нескольких интерфейсах ПАК, то результат синхронизации времени на самом ПАК и на NTP-клиентах не определен, так как неизвестно, какой интерфейс будет включен первым и какой список дополнительных NTP-серверов получит ПАК.

Настройка и управление NTP-сервером осуществляется с помощью команд из группы `inet` (см. «[Команды подгруппы inet ntp](#)» на стр. 75). Управление сводится к запуску и остановке сервера. Запуск NTP-сервера производится командой `inet ntp start`, для остановки сервера используется команда `inet ntp stop`. Кроме того, можно настроить автоматический запуск NTP-сервера при старте ПАК с помощью команд `inet ntp mode on` (включить автоматический запуск) и `inet ntp mode off` (выключить автоматический запуск).



Примечание. При организации кластера в условиях ограничений по выделению IP-адресов при переходе сервера из пассивного режима в активный NTP-сервер будет перезапускаться автоматически, если он был отмечен к запуску (`inet ntp mode on`), вне зависимости от того, был ли он до этого запущен или нет.

Изменение настройки автоматического запуска не влияет на текущее состояние NTP-сервера, новая настройка вступает в силу только при следующем старте ПАК. Для изменения состояния NTP-сервера в текущем сеансе работы следует использовать команды `inet ntp start` и `inet ntp stop`. Автоматический запуск NTP-сервера по умолчанию включен, и при первом после установки ПО старте ПАК NTP-сервер запускается.



Внимание! Если автоматический запуск локального NTP-сервера включен, но при старте ПАК ни один из NTP-серверов, указанных в его настройках, недоступен, то локальный NTP-сервер не будет запущен, а на консоли появится соответствующее предупреждение.

Перед каждым запуском NTP-сервера на ПАК запускается NTP-клиент для синхронизации времени на самом ПАК. NTP-сервер будет запущен только в случае успешной синхронизации времени на ПАК. Если NTP-клиенту не удалось синхронизировать время на ПАК, то NTP-сервер не запускается.

Формирование списка NTP-серверов, используемых для синхронизации, осуществляется с помощью команд добавления (`inet ntp add`) и удаления (`inet ntp delete`). Для просмотра списка NTP-серверов используется команда `inet ntp list`.

Если на ПАК запущен DHCP-сервер, то в качестве адреса NTP-сервера он всегда предоставляет DHCP-клиентам свой собственный адрес.

Для просмотра текущего состояния NTP-сервера и его настроек используется команда `inet show ntp` (см. «[Команды группы inet](#)» на стр. 67). Данная команда также позволяет получить информацию о функционировании NTP-серверов, с которыми взаимодействует локальный NTP-сервер. Эта информация помогает выявлять и устранять проблемы,

связанные с настройкой времени. Например, когда клиенты, использующие ПАК в качестве NTP-сервера, перестают синхронизировать свое время, необходимо проверить уровень (stratum) сервера, который является текущим источником синхронизации (этот сервер отмечен звездочкой). Если у текущего сервера слишком высокий уровень, его следует удалить из настроек локального NTP-сервера. У других серверов (прежде всего тех, которые отмечены знаком плюс) также необходимо проверить уровень и параметры, отражающие надежность серверов (например, параметры reach и jitter). Недостаточно надежные серверы рекомендуется удалить из настроек. Целесообразно также удалить из настроек NTP-серверы, отмеченные знаком 'x' или '-'.

Взаимодействие ПАК с UPS

В состав ПАК ViPNet Coordinator HW входит пакет Network UPS Tools (NUT), предназначенный для организации взаимодействия ПАК с источником бесперебойного питания (UPS, Uninterruptable Power Supply).

При подключении ПАК к UPS последний обеспечивает работу компьютера только до тех пор, пока не разрядится батарея. После разряда батареи компьютер будет некорректно выключен, что может привести к потере данных. В то же время большинство современных UPS могут подключаться к компьютеру с помощью интерфейсного кабеля и посылать сигнал об истощении батареи. Полученный сигнал компьютер может использовать для корректного выключения.

Взаимодействие компьютера с UPS обеспечивает пакет NUT, который включает в себя:

- набор драйверов различных производителей UPS;
- демон upsd, взаимодействующий с драйвером;
- демон upsmon, взаимодействующий с демоном upsd; именно этот демон осуществляет мониторинг состояния UPS и при необходимости производит корректное выключение компьютера.



Примечание. В настоящее время поддерживаются только UPS фирмы APC, подключаемые через USB-порт.

В общем случае от одного UPS могут питаться несколько компьютеров. Компьютер, к которому подключен интерфейсный кабель UPS, является главным (master) и отвечает за своевременное оповещение по сети подчиненных (slave) компьютеров. Драйвер UPS и демон upsd работают только на главном компьютере (master); демон upsmon работает на всех компьютерах и взаимодействует с демоном upsd локально или удаленно (см. Рисунок 32 на стр. 197).



Рисунок 32: Схема взаимодействия ПАК с UPS

Примечание. Для обеспечения резервирования электропитания модуля VPNM в модификации ПАК ViPNet Coordinator HW-VPNM необходимо интерфейсный кабель UPS подключить к модулю, а кабель электропитания — к маршрутизатору. При таком подключении будет обеспечена возможность корректного выключения модуля по сигналу от UPS. Возможность резервирования электропитания непосредственно маршрутизатора следует решать средствами и способами, рекомендованными производителем маршрутизатора.

Для настройки и управления взаимодействием ПАК с UPS предназначены команды группы `ups` (см. «[Команды группы ups](#)» на стр. 84). С помощью этих команд можно:

- включить или отключить взаимодействие ПАК с UPS;
- задать режим взаимодействия ПАК с UPS (`master` или `slave`);
- запустить или остановить демоны пакета NUT;
- просмотреть текущие настройки взаимодействия ПАК с UPS;
- просмотреть текущее состояние UPS.

Чтобы настроить взаимодействие с UPS, последовательно выполните следующие действия:

- 1 На ПАК, который будет выступать в роли master:
 - 1.1 Подключите интерфейсный кабель UPS к ПАК, который будет выступать в роли master.
 - 1.1 Включите службу UPS (командой `ups set monitoring on`).
 - 1.2 Проверьте, установлен ли на ПАК режим взаимодействия master (командой `ups show config`). Режим master устанавливается по умолчанию. При необходимости установите данный режим (командой `ups set mode master`).
 - 1.3 Запустите службы UPS (командой `ups start`).
 - 1.4 Проверьте взаимодействие службы с UPS (командой `ups show status`). Если взаимодействие установлено, выведется статистика с UPS.
- 2 На ПАК, который будет выступать в роли slave:
 - 2.1 Включите службу UPS (командой `ups set monitoring on`).



Внимание! При подключении к UPS кластера горячего резервирования никакие другие компьютеры к UPS подключать нельзя.

- 2.2 Установите на ПАК режим взаимодействия slave (командой `ups set mode slave <master_IP>`). ПАК, выступающий в роли master, должен быть доступен с текущего ПАК по этому IP-адресу.

При подключении к UPS кластера горячего резервирования в качестве IP-адреса мастера укажите адрес первого ПАК на резервном канале.

- 2.3 Запустите службы UPS (командой `ups start`).
- 2.4 Проверьте взаимодействие службы с UPS с master (командой `ups show status`). Если взаимодействие установлено, выведется статистика с UPS, подключенного к master.



Примечание. При подключении к UPS кластера горячего резервирования для связи между компьютерами можно использовать только резервный канал, на котором IP-адреса неизменны при переходе ПАК из активного режима кластера в пассивный и наоборот.

Для автоматического включения ПАК после появления питания в BIOS необходимо настроить параметры, приведенные в таблице ниже.

Таблица 10. Параметры настройки BIOS для автоматического включения ПАК после появления питания

Модификация ПАК	Пункт меню/подменю	Параметр	Значение
ПАК ViPNet Coordinator HW1000 G1/G2	Power > APM Configuration	Restore on AC Power Loss	Power On
ПАК ViPNet Coordinator HW100 G1	Advanced > APM Configuration	Restore on AC/Power Loss	Power On
ПАК ViPNet Coordinator HW100 G2	Integrated Peripherals > SuperIO Device	PWRON After PWR-Fail	On



15

Протоколирование событий, ведение и просмотр журналов

Журнал регистрации IP-пакетов	201
Журнал транспортных конвертов MFTR	208
Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP	210
Журналы устранения неполадок ПО ViPNet	213
Экспорт журналов устранения неполадок ПО ViPNet	216

Журнал регистрации IP-пакетов

Журнал регистрации IP-пакетов в драйвере сетевой защиты `iplir` ведется отдельно по каждому адаптеру. События, относящиеся к защищенной сети, генерируются подсистемой драйвера `iplir`, ответственной за работу защищенной сети, а события, относящиеся к открытой сети – подсистемой обработки открытых пакетов. При этом и те, и другие помещаются в одну и ту же очередь для данного адаптера, из которой их затем забирает управляющий демон, который сохраняет данные журнала в базе данных на жестком диске.

В журнал заносится информация не о соединениях, а о пакетах, которые проходят через данный адаптер. Таким образом, все пропущенные транзитные пакеты отображаются в журнале дважды: первый раз – на интерфейсе, на который они приходят, второй раз – на интерфейсе, через который они уходят. Все пропущенные локальные пакеты отображаются один раз – на том интерфейсе, через который они приходят или уходят. Все заблокированные пакеты отображаются в журнале один раз – на том интерфейсе, на котором они появились и были заблокированы. Это относится и к незашифрованным пакетам, и к зашифрованным.

Просмотр журнала регистрации IP-пакетов осуществляется с помощью команды `iplir view` (см. «Команды группы `iplir`» на стр. 76). При просмотре журнала предоставляется возможность вывести события, отобранные по следующим параметрам:

- интервал дат;
- сетевой интерфейс, на котором был обработан пакет;
- IP-протокол;
- направление пакета – входящий или исходящий;
- тип события;
- диапазон или одно значение IP-адреса отправителя и/или получателя пакета;
- диапазон или одно значение местного порта для TCP, UDP;
- диапазон или одно значение удаленного порта для TCP, UDP;
- имя пользователя защищенной сети – отправителя и/или получателя пакета.

После ввода команды `iplir view` экран терминала принимает следующий вид:

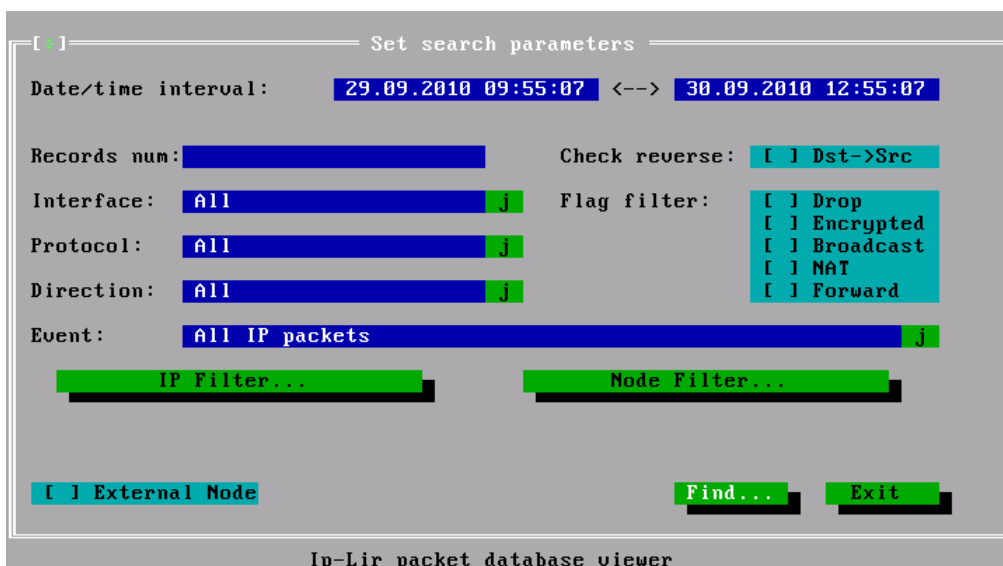


Рисунок 33: Задание параметров поиска в журнале регистрации IP-пакетов

По умолчанию предполагается просмотр собственного журнала. Если необходимо просмотреть журнал удаленного узла, надо включить опцию **External Node**, после чего появится запрос пароля администратора сети ViPNet. Если введен правильный пароль, справа от опции **External Node** появится кнопка **Select**. По этой кнопке открывается окно со списком защищенных узлов, в котором надо выбрать нужный узел. Для успешного подключения к удаленному узлу необходимо, чтобы узел был доступен на уровне TCP/IP и на нем работала программа управления. Если подключиться не удастся, программа сообщает об этом и завершает свою работу.

Для поиска записей в журнале регистрации IP-пакетов можно задать следующие параметры:

- **Date/time interval** – интервал дат и времени, в котором будет производиться поиск записей о регистрации пакетов, в формате ДД.ММ.ГГГГ ЧЧ:ММ:СС.
- **Records num** – количество выводимых записей.
- **Interface** – сетевой интерфейс (выбирается из списка доступных интерфейсов).
Доступным считается интерфейс, у которого существует журнал IP-пакетов. Поиск пакетов будет производиться в журнале выбранного интерфейса. В качестве параметра может быть указано имя интерфейса или значение **All** для работы со всеми интерфейсами.
- **Protocol** – протокол для поиска среди всех пакетов только пакетов по заданному IP-протоколу.

В качестве параметра может быть указано имя протокола или значение **All** для работы со всеми протоколами.

- **Direction** – направление прохождения пакета, может принимать одно из следующих значений:
 - **All** – входящие и исходящие пакеты;
 - **Incoming** – входящие пакеты;
 - **Outgoing** – исходящие пакеты.

- **Check reverse** – признак включения в журнал ответных пакетов от получателя отправителю.

Имеет смысл при указании конкретного IP-адреса (**IP Filter**) или узла (**Node Filter**) в качестве отправителя и/или получателя пакетов.

- **Flag filter** – признаки для поиска среди всех пакетов только пакетов с одним или несколькими из заданных признаков:
 - **Drop** – заблокированные пакеты;
 - **Encrypted** – зашифрованные пакеты;
 - **Broadcast** – широковещательные пакеты;
 - **NAT** – транслированные пакеты;
 - **Forward** – транзитные пакеты.

- **Event** – событие для поиска среди всех пакетов только пакетов с заданным событием.

Выбирается из списка, по умолчанию поиск производится среди всех событий.

- **IP Filter** – показывает окно для задания следующих параметров запроса:
 - **Source IP address** – **All**, диапазон или одиночное значение для допустимого IP-адреса отправителя пакета;
 - **Destination IP address** – **All**, диапазон или одиночное значение для допустимого IP-адреса получателя пакета;
 - **Source port** – диапазон или одиночное значение для допустимого номера порта отправителя (0-65535) для протоколов TCP, UDP;
 - **Destination port** – диапазон или одиночное значение для допустимого номера порта получателя (0-65535) для протоколов TCP, UDP.
- **Node Filter** – показывает окно для задания параметров запроса по узлу отправителя и/или получателя: **Source** и/или **Destination**.

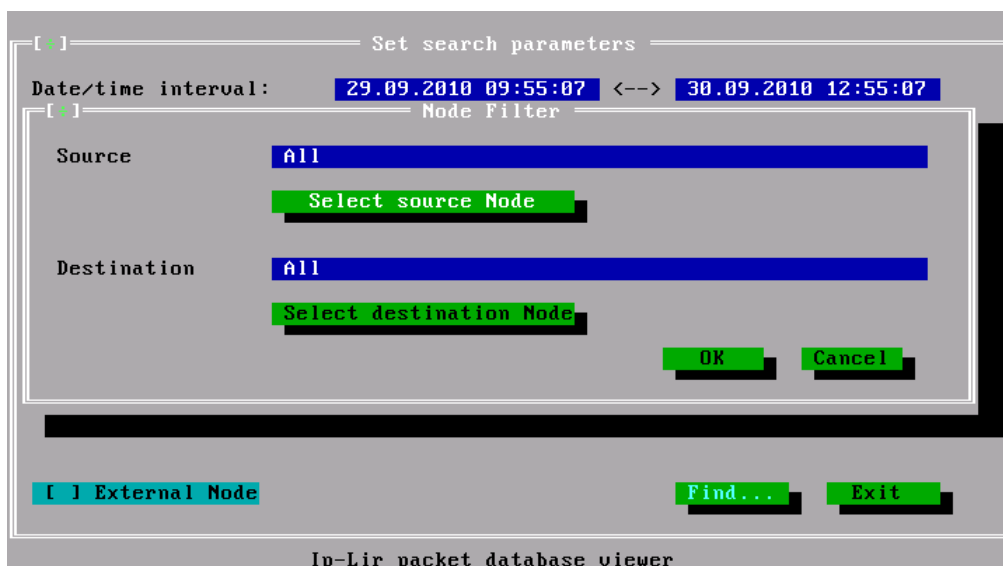


Рисунок 34: Задание параметров запроса по узлу отправителя и/или получателя

Для непосредственного выбора узла отправителя или получателя необходимо использовать соответствующие кнопки: **Select source Node** или **Select destination Node**. При этом открывается окно со списком защищенных узлов и возможностью поиска в этом списке.

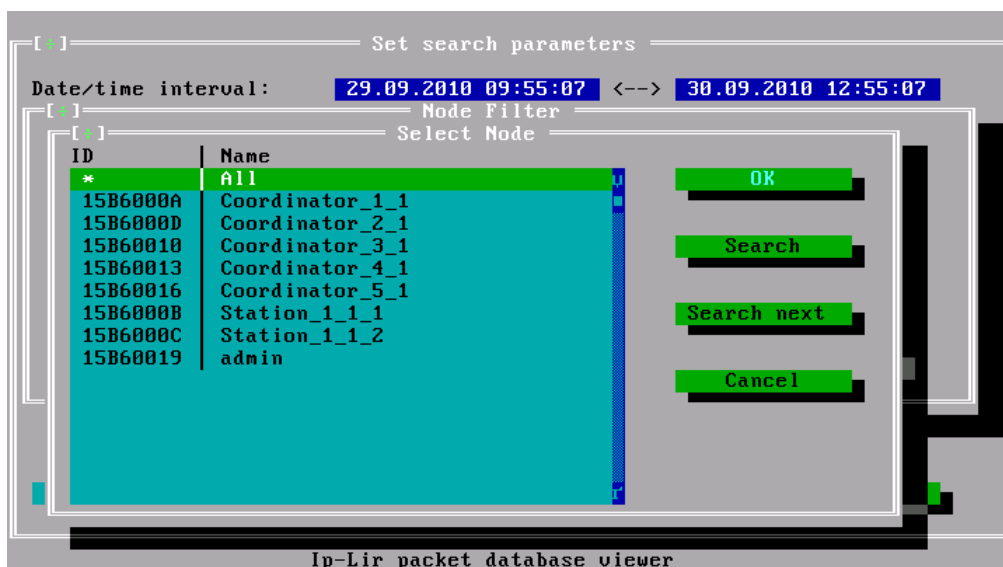


Рисунок 35: Список защищенных узлов для выбора отправителя или получателя

Окно выбора узлов содержит список всех защищенных узлов ViPNet, связанных с данным узлом. Узлы в списке отсортированы в алфавитном порядке. В левом столбце отображается шестнадцатеричный идентификатор узла. Помимо узлов, список содержит служебный элемент **All**, выбор которого соответствует фильтрации по всем узлам.

Для поиска узлов нажмите кнопку **Search**. При этом в отдельном окне отображается подстрока поиска с возможностью ручного ввода и выбора ранее введенной подстроки из выпадающего списка. В качестве критерия поиска можно использовать как имя узла, так и уникальный идентификатор узла, то есть поиск будет осуществляться на соответствие вхождения введенной подстроки как в **Name**, так и в **ID**.

Кнопка **Search next** предназначена для быстрого поиска следующего элемента списка, удовлетворяющего ранее выбранной подстроке поиска.

Для выполнения запроса журнала по заданным параметрам нажмите кнопку **Find**, расположенную в нижней части основного окна (см. Рисунок 33 на стр. 202). Для завершения просмотра журнала нажмите кнопку **Exit**.

Список записей, найденных в журнале регистрации IP-пакетов, выводится в окне **View results** (см. Рисунок 36 на стр. 206). Записи упорядочены по времени регистрации пакета. Список состоит из следующих колонок:

- Дата и время регистрации события.
- Интерфейс, на котором зарегистрировано событие.
- Направление движения зарегистрированного пакета: «<» исходящие, «>» входящие и флаги события:
 - **C** – шифрованный пакет;
 - **B** – широковещательный пакет;
 - **D** – заблокированный пакет;
 - **T** – транзитный (маршрутизируемый) пакет;
 - **R** – пакет будет обработан правилами NAT открытой сети;
 - **N** – пакет был обработан правилами NAT открытой сети.
- Протокол.
- IP-адрес источника.
- Местный порт для протоколов TCP и UDP.
- IP-адрес назначения.
- Порт удаленного компьютера для протоколов TCP и UDP.

В нижней части окна отображается информация по выбранному в списке событию:

- Название события, присвоенного IP-пакету.
- Интерфейс.
- Протокол.
- Размер пакета.

Показывается суммарный размер всех пакетов, относящихся к данному событию, если счетчик больше единицы. Для зашифрованных пакетов показывается полный размер пакетов со всеми служебными заголовками, необходимыми для работы защищенной сети.

- Число пакетов, относящихся к данному событию.
- Узел отправителя.
- Узел получателя.

Date/time	Dev	Flags	Prot	Source IP	Port	Destination IP	Port
09/29 11:04:41	eth1	>-----	udp	192.168.2.200	67	192.168.2.14	68
09/29 11:04:41	eth1	<-C---	udp	192.168.2.14	2046	192.168.4.15	2046
09/29 11:04:41	eth1	<-C---	udp	192.168.2.14	2046	192.168.4.5	2046
09/29 11:04:41	eth1	<-C---	udp	192.168.2.14	2046	160.0.9.15	2046
09/29 11:04:41	eth1	<-C---	udp	192.168.2.14	2046	1.0.7.5	2046
09/29 11:04:41	eth1	<-C---	udp	192.168.2.14	68	192.168.2.200	67
09/29 11:04:41	eth1	<-----	udp	192.168.2.14	68	192.168.2.200	67
09/29 11:04:41	eth0	>D---T	udp	192.168.1.11	32768	198.32.64.12	53
09/29 11:04:40	eth0	>D---T	udp	192.168.1.11	32768	193.0.14.129	53
09/29 11:04:38	eth0	>D---T	udp	192.168.1.11	32768	128.63.2.53	53
09/29 11:04:37	eth1	>-C---	icmp	192.168.2.3	0	192.168.1.11	0
09/29 11:04:37	eth1	<-C---	icmp	192.168.2.14	0	192.168.2.3	0
09/29 11:04:37	eth0	>D---T	udp	192.168.1.11	32768	192.5.5.241	53

40 - Encrypted IP packet allowed

Interface : eth1 Packets Size : 1098 Total In : 944 Kb
Eth. proto: 800h Packets Count: 6 Total Out: 955 Kb

Source Node: (15B6000A) Coordinator_1_1
Destin Node: (15B60013) Coordinator_4_1

Esc - return to main window Enter - view details F2 - export to file

Рисунок 36: Вывод списка найденных записей



Внимание! Функция экспорта журнала в файл, вызываемая по клавише **F2** (export to file), на ПАК ViPNet Coordinator HW не работает!

Для любой выбранной в списке записи можно просмотреть более детальную информацию, для этого нажмите клавишу **Enter**.

```
[ ] Record details
Events: 40 - Encrypted IP packet allowed

Interval Begin: 29.09.2010 11:04:41
                End: 29.09.2010 11:05:11

Interface: eth1      Ethernet protocol: 800h
Size:      1098      Count:              6

Drop:      NO        Encrypted YES
Direction: Outgoing  NAT:              NO
Broadcast: NO        Forward:          NO

IP protocol: 17 - UDP (User Datagram)
Source IP:   192.168.2.14      Port: 2046
Destination IP: 192.168.4.5    Port: 2046

Key number:          FFFFFFFE
Source Node          15B6000A
  Coordinator_1_1
Destination Node     15B60013
  Coordinator_4_1

Esc or Enter - return to view results
```

Рисунок 37: Детальная информация о событии

В нижней части окна со списком найденных в журнале записей (см. Рисунок 36 на стр. 206) расположены поля **Total In** и **Total Out**, в которые помещается информация о суммарном размере входящих и исходящих пакетов соответственно. Суммарный размер считается для всех пакетов, которые были получены в результате запроса. Если для каких-либо записей журнала пакетов информации о размере пакетов нет, то эти записи не учитываются при подсчете объема трафика. В этом случае после размера в соответствующем поле (**Total In** и/или **Total Out**) отображается звездочка – признак того, что не весь трафик учтен. Если же во всех найденных записях нет информации о размере пакетов, то в соответствующем поле (**Total In** и/или **Total Out**) отображается **N/A** (звездочка не отображается).

Единицы измерения при отображении суммарного размера выбираются следующим образом:

- если суммарный размер меньше 100 килобайт, то размер отображается в байтах и к числу добавляется суффикс «b»;
- если суммарный размер больше 100 килобайт, но меньше 100 мегабайт, то размер отображается в килобайтах и к числу добавляется суффикс «Kb»;
- если суммарный размер больше 100 мегабайт, то размер отображается в мегабайтах и к числу добавляется суффикс «Mb».

Журнал транспортных конвертов

MFTP

В процессе работы транспортный модуль осуществляет запись информации об обработанных конвертах в специальную базу данных (БД), называемую журналом конвертов. В журнал заносится следующая информация:

- о полностью принятых конвертах;
- об отправленных конвертах;
- об удаленных конвертах;
- о поврежденных конвертах.

Алгоритм работы с БД журнала основан на использовании ротации. В настройках транспортного модуля задается максимальный размер файла журнала в мегабайтах. При превышении указанного размера осуществляется запись поверх самых старых записей. Таким образом, задавая максимальный размер файла журнала, можно регулировать необходимое число хранимых записей на текущий момент. При изменении максимального размера журнала в процессе работы происходит реконструкция файла БД. В случае уменьшения размера (по сравнению с предыдущим значением) теряются записи с наиболее старыми датами.

При экспорте информации из журнала в текстовый файл выводятся значения следующих полей:

- имя файла конверта;
- размер конверта;
- имя узла-отправителя;
- имя узла-получателя;
- имя события;
- имя прикладной задачи отправителя;
- описание конверта;

- дата и время события.

Просмотр журнала транспортных конвертов осуществляется с помощью команды `mftp view` (см. «[Команды группы mftp](#)» на стр. 79).

Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP

ПО ViPNet в процессе работы взаимодействует с сервером SNMP, что позволяет удаленно получать информацию о времени работы системы, интерфейсах и их режимах, количестве пропущенных и заблокированных пакетов и т.д., а также уведомлять удаленную станцию сетевого менеджмента (NMS) о наиболее важных событиях в системе. Сервер SNMP запускается автоматически при старте ПО ПАК ViPNet Coordinator HW.

Если планируется использовать оповещения (SNMP Traps), то необходимо настроить адрес удаленного узла, на который будут отправляться асинхронные оповещения о различных событиях при работе ПО ViPNet. Данная настройка осуществляется командой `machine set snmp-trapsink` (см. «Команды группы machine» на стр. 82).

Для получения информации по протоколу SNMP используется специальное ПО сетевого менеджмента (NMS), например HP OpenView. Перед началом работы с данным узлом необходимо импортировать в NMS специальный MIB-файл ViPNet, который описывает используемые ViPNet объекты SNMP. Этот файл называется `VIPNET-MIB.txt`. Действия, которые нужно проделать для такого импорта, зависят от используемой NMS и должны быть описаны в документации на нее.

После импорта MIB-файла с данной NMS можно получать информацию о состоянии ПО ViPNet на узле. Для этого должна использоваться ветка

```
.iso.org.dod.internet.private.enterprises.infotecs.vipnet  
(.1.3.6.1.4.1.10812.1)
```

ПО ViPNet использует следующие объекты (для всех объектов возможно только чтение данных):

- .1.3.6.1.4.1.10812.1.1 – сетевой идентификатор ViPNet для данного узла;
- .1.3.6.1.4.1.10812.1.1.2 – название данного узла;
- .1.3.6.1.4.1.10812.1.1.3 – имя пользователя, пароль которого был введен для запуска ViPNet;
- .1.3.6.1.4.1.10812.1.1.4 – время работы системы;

- .1.3.6.1.4.1.10812.1.2.1 – число сетевых интерфейсов в системе;
- .1.3.6.1.4.1.10812.1.2.2 – последовательность (sequence) объектов, описывающих сетевые интерфейсы;
- .1.3.6.1.4.1.10812.1.2.2.1 – каждый из объектов, описывающих сетевые интерфейсы. В него входят следующие поля:
 - .1.3.6.1.4.1.10812.1.2.2.1.1 – номер интерфейса;
 - .1.3.6.1.4.1.10812.1.2.2.1.2 – системное имя интерфейса;
 - .1.3.6.1.4.1.10812.1.2.2.1.3 – режим работы интерфейса в ViPNet;
 - .1.3.6.1.4.1.10812.1.2.2.1.4 – число пропущенных входящих нешифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.5 – число заблокированных входящих нешифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.6 – число пропущенных исходящих нешифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.7 – число заблокированных исходящих нешифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.8 – число пропущенных входящих зашифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.9 – число заблокированных входящих зашифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.10 – число пропущенных исходящих зашифрованных пакетов;
 - .1.3.6.1.4.1.10812.1.2.2.1.11 – число заблокированных исходящих зашифрованных пакетов.

ПО ViPNet использует следующие оповещения (SNMP Traps):

- .1.3.6.1.4.1.10812.1.3.1 – запуск управляющего демона;
- .1.3.6.1.4.1.10812.1.3.2 – остановка управляющего демона;
- .1.3.6.1.4.1.10812.1.3.3 – запуск демона mftpd;
- .1.3.6.1.4.1.10812.1.3.4 – остановка демона mftpd;
- .1.3.6.1.4.1.10812.1.3.7 – запуск демона failoverd (только в активном режиме);

- .1.3.6.1.4.1.10812.1.3.8 – остановка демона failoverd (только в активном режиме);
- .1.3.6.1.4.1.10812.1.3.9 – переключение демона failoverd из пассивного режима в активный.

Как правило, NMS позволяет настроить определенную реакцию на каждое из оповещений (отсылка e-mail, sms и т.п.).

Журналы устранения неполадок ПО ViPNet

В процессе работы ПАК службы (демоны), входящие в состав ПО ViPNet, ведут журналы (логи), предназначенные для протоколирования и диагностики работы соответствующей службы. Данные журналы позволяют диагностировать правильность функционирования ПО, а также выявлять неполадки в работе служб ViPNet. Информация, содержащаяся в журналах, особенно на высоком уровне протоколирования, содержит большое количество деталей о процессах, происходящих внутри служб ViPNet, и предназначена для разработчиков. Эта информация наряду с дампами аварийного завершения программ используется для исследования и выработки рекомендаций по устранению неполадок функционирования ПО ViPNet.

Запись сообщений в журналы устранения неполадок осуществляется системными средствами с использованием протокола syslog. Журналы могут храниться двумя способами:

- В системном журнале, расположенном на жестком диске ПАК (локальное протоколирование).
- В системном журнале, расположенном на удаленном syslog-сервере (удаленное протоколирование).



Внимание! Локальное протоколирование возможно только на модификациях ПАК с жестким диском: ПАК ViPNet Coordinator HW100 при комплектации компьютера жестким диском, ПАК ViPNet Coordinator HW1000, ПАК ViPNet Coordinator HW-VPNM.

Способ протоколирования устанавливается с помощью команды `machine set loghost` (см. «Команды группы `machine`» на стр. 82), в которой указывается либо значение `local` для хранения журналов на ПАК, либо адрес удаленного syslog-сервера. При локальном протоколировании журналы можно посмотреть с помощью команды `machine show logs`.



Примечание. На модификациях ПАК с жестким диском по умолчанию установлен локальный способ протоколирования.

В случае хранения файла с журналами на ПАК предусмотрена автоматическая ротация файла, чтобы предотвратить критическое уменьшение свободного места на жестком диске. Ротация выполняется при достижении файлом размера 1 ГБ, при этом предыдущие файлы с журналами переименовываются и самый старый файл удаляется с тем, чтобы на диске осталось не более 3-х предыдущих файлов. Кроме того, можно вручную удалить файл (файлы) с журналами с помощью команды `admin remove logs` (см. «Команды группы `admin`» на стр. 79).

Настройка протоколирования осуществляется с помощью параметров, задаваемых в секции `[debug]` конфигурационных файлов демонов. В секции `[debug]` задаются следующие параметры:

- `debuglevel` – уровень протоколирования;
- `debuglogfile` – источник (`facility`) и важность (`level`) для сообщений в системном журнале, задается в виде `syslog:<facility.level>`.

Уровень протоколирования определяет степень детализации формируемых сообщений и задается числом от -1 до 5. С повышением уровня протоколирования увеличивается количество информации, присутствующей в сообщениях. Для отключения протоколирования используется значение уровня, равное -1. По умолчанию уровень протоколирования равен 3.

Источник сообщений (`facility`) определяет процесс, который генерирует сообщения. Источник сообщений может принимать, в частности, следующие значения:

- `kern` (ядро);
- `user` (пользовательские программы);
- `mail` (почтовая система);
- `daemon` (демоны).

Важность сообщений (`level`) определяет уровень серьезности сообщений и может принимать, в частности, следующие значения:

- `err` (ошибка);
- `info` (информационное сообщение);
- `debug` (отладочное сообщение).

По умолчанию `facility` устанавливается в значение `daemon`, `level` – в значение `debug`.

Сообщения о работе каждого из демонов могут быть выведены на консоль командного интерпретатора. Для этого необходимо выполнить следующее:

- 1 В конфигурационном файле контролируемого демона в секции [debug] задать уровень протоколирования и требуемые facility и level.
- 2 Включить вывод сообщений на консоль командного интерпретатора с помощью команды `debug on` (см. «Прочие команды» на стр. 85), указав в параметрах те же facility и level.

Для отключения вывода сообщений на консоль интерпретатора используется команда `debug off` (см. «Прочие команды» на стр. 85). С помощью этой команды можно отключить как вывод всех сообщений, так и вывод только тех сообщений, которые соответствуют конкретным facility и level.

Например, можно настроить протоколирование таким образом, чтобы следить за работой разных демонов на разных терминалах. Для этого надо выполнить следующее:

- 1 В конфигурационном файле одного из демонов в секции [debug] задать нужные параметры, например:

```
[debug]
debuglogfile= syslog:daemon.debug
debuglevel= 3
```

и на одном из терминалов выполнить команду `debug on daemon.debug`.

- 2 В конфигурационном файле другого демона в секции [debug] задать параметры протоколирования, указав другое значение facility, например:

```
[debug]
debuglogfile= syslog:local0.debug
debuglevel= 3
```

и на другом терминале выполнить команду `debug on local0.debug`.

При таких настройках на первый терминал будут поступать сообщения только от первого демона, на второй терминал – только от второго демона. Чтобы на первом терминале можно было следить за работой обоих демонов, на нем в дополнение к первой команде надо выполнить команду `debug on local0.debug`.

Выводимые на консоль командного интерпретатора сообщения о работе демонов могут смешиваться с сообщениями самого интерпретатора и набираемой на консоли командой. Для просмотра набранной к данному моменту части команды и продолжения ее ввода используется комбинация клавиш **<Ctrl+L>**.

Экспорт журналов устранения неполадок ПО ViPNet

Журналы устранения неполадок ПО ViPNet, которые ведутся локально на ПАК с жестким диском, можно экспортировать (выгрузить) с ПАК на другой компьютер (ноутбук) или на USB-флэш. Для выполнения экспорта на ноутбуке должна быть установлена сетевая карта Ethernet и ОС Windows XP или Windows Vista, флэш-память должна быть отформатирована в одну из поддерживаемых файловых систем: FAT32 или ext2. Журналы экспортируются в архивированном виде в файл с именем `logs.tar.gz`.

Экспорт на компьютер



Внимание! Во время выполнения экспорта журналов на другой компьютер будут остановлены все демоны и выгружены все драйверы ViPNet, т.е. ПАК окажется незащищенным!

При экспорте журналов устранения неполадок с ПАК на другой компьютер используется стандартная служба TFTP. В ОС Windows XP эта служба по умолчанию включена. В ОС Windows Vista эта служба по умолчанию отключена и ее необходимо включить вручную. Для включения службы в ОС Windows Vista выполните следующее:

- 1 Выберите **Start > Control Panel > Programs and Features**.
- 2 Зайдите в меню **Turn Windows features on or off** и включите службы **TFTP Client** и **Simple TCP/IP services**.

Экспорт выполняется в следующей последовательности:

- 1 Подключите ноутбук к порту Ethernet1 ПАК с помощью кроссированного кабеля Ethernet.
- 2 Установите вручную на сетевом интерфейсе ноутбука IP-адрес 169.254.241.5.
- 3 Подключите к ПАК СОМ-консоль или обычную консоль (монитор и клавиатуру).
- 4 Остановите все демоны с помощью соответствующих команд.

- 5 Выполните команду `admin export logs tftp` (см. «Команды группы admin» на стр. 79).

При выполнении команды производится архивирование журналов устранения неполадок. Можно прервать архивирование, нажав клавиши `<Ctrl+C>`. В случае прерывания архивирования экспорт прекращается.

После завершения архивирования появляется сообщение, содержащее имя файла с архивом, и предложение скачать этот файл.

- 6 Перенесите файл с архивом на ноутбук по TFTP с помощью следующей команды:

```
tftp -i 169.254.241.1 get <имя файла>
```

- 7 Нажмите ввод.

После завершения экспорта или в случае прерывания архивирования необходимо запустить все демоны с помощью соответствующих команд.

Экспорт на USB-флэш

Экспорт журналов устранения неполадок на USB-флэш выполняется в следующей последовательности:

- 1 Подключите к ПАК СОМ-консоль или обычную консоль (монитор и клавиатуру).
- 2 Выполните команду `admin export logs usb` (см. «Команды группы admin» на стр. 79).

При выполнении команды производится архивирование журналов устранения неполадок. Можно прервать архивирование, нажав клавиши `<Ctrl+C>`. В случае прерывания архивирования экспорт прекращается.

После завершения архивирования появляется предложение вставить USB-флэш и подтвердить продолжение экспорта. В случае отказа от продолжения экспорт прекращается.

- 3 Вставьте USB-флэш в USB-разъем ПАК.

Если объем свободного места на USB-флэш меньше, чем размер файла с архивом, появляется соответствующее сообщение и повторное предложение вставить USB-флэш.

Если свободного места на USB-флэш достаточно, файл с архивом копируется на USB-флэш. Можно прервать копирование, нажав клавиши `<Ctrl+C>`. В случае прерывания копирования экспорт прекращается. После завершения копирования появляется сообщение об окончании экспорта.



16

Обновление ПО ПАК ViPNet Coordinator HW

Удаленное обновление ПО	219
Локальное обновление ПО	220

Удаленное обновление ПО

Удаленное обновление ПО ПАК ViPNet Coordinator HW производится с помощью программного обеспечения ViPNet Administrator Центр управления сетью (ЦУС). Процедура удаленного обновления ПО подробно описана в документе «ViPNet Administrator Центр управления сетью. Руководство администратора».

С помощью удаленного обновления на ПАК обновляются все компоненты ПО.

Примечание. Для разных модификаций ПАК файл дистрибутива обновления ПО называется по-разному:



- hw100_vipnet_<Major>.<Minor>-<Build>.lzh для ПАК ViPNet Coordinator HW100;
- hw1000_vipnet_<Major>.<Minor>-<Build>.lzh для ПАК ViPNet Coordinator HW1000;
- hwvpm_vipnet_<Major>.<Minor>-<Build>.lzh для ПАК ViPNet Coordinator HW-VPNМ.

Для любой модификации ПАК перед отправкой дистрибутива обновления ПО необходимо переименовать файл дистрибутива в `driv.lzh`.

Локальное обновление ПО

Локальное обновление ПО ПАК ViPNet Coordinator HW производится с помощью команды `admin upgrade software usb`. Для обновления необходимо наличие USB-флэш, на которую предварительно записан файл дистрибутива обновления ПО (файл `driv.lzh`).



Примечание. На одну USB-флэш можно записать несколько дистрибутивов, предназначенных для обновления ПО на различных модификациях ПАК (разместив их в разных директориях). При выполнении команды локального обновления производится анализ содержимого USB-флэш и выбор только тех дистрибутивов, которые соответствуют данной модификации ПАК.

Локальное обновление ПО выполняется в следующей последовательности:

- 1 Подключите к ПАК COM-консоль или обычную консоль (монитор и клавиатуру).
- 2 Выполните команду `admin upgrade software usb`.
При старте команды появляется предложение вставить USB-флэш.
- 3 Вставьте USB-флэш в USB-разъем ПАК и подтвердите это.
Если USB-флэш не найдена, появляется соответствующее сообщение и выполнение команды завершается, обновление не производится.
- 4 На USB-флэш производится поиск файлов `driv.lzh`, соответствующих данной модификации ПАК:
 - если файлов нет, появляется соответствующее сообщение и выполнение команды завершается, обновление не производится;
 - если файлы найдены, выводится пронумерованный список директорий, в которых найдены файлы обновлений, и предлагается ввести номер нужной директории либо отказаться от обновления. В случае отказа выполнение команды завершается, обновление не производится.
- 5 Введите номер директории, затем нажмите ввод.
- 6 Выполняется обновление ПО из выбранного файла.
Если обновление производится на ПАК, входящем в состав кластера горячего резервирования, появляется напоминание о необходимости обновления ПО на втором ПАК кластера.



Внимание! При наличии кластера горячего резервирования локальное обновление ПО должно быть произведено на обоих ПАК кластера.

После завершения обновления появляется сообщение об успешном или неуспешном результате обновления. В случае успешного обновления необходимо перезагрузить ПАК, чтобы обновление вступило в силу.

Процесс локального обновления ПО на кластере горячего резервирования имеет ряд особенностей и подробно описан в документе «ПАК ViPNet Coordinator HW. Система защиты от сбоев. Руководство администратора».



17

Настройка работы ПАК ViPNet Coordinator HW-VPNМ

Режимы работы маршрутизатора Huawei Secoway USG	223
Настройка при работе маршрутизатора в режиме transparent	225
Настройка при работе маршрутизатора в режиме router	231

Режимы работы маршрутизатора Huawei Secoway USG

Маршрутизатор Huawei Secoway USG имеет два режима работы:

- Режим `transparent`, в котором маршрутизатор работает в качестве сетевого коммутатора.

В этом режиме маршрутизатор прозрачно пропускает трафик от компьютеров, подключенных со стороны маршрутизатора, до модуля VPNM. Пример схемы сетевых подключений и настройки работы в этом режиме приведен в разделе [Настройка при работе маршрутизатора в режиме `transparent`](#) (на стр. 225).

- Режим `router`, в котором маршрутизатор работает в качестве межсетевого экрана.

В этом режиме доступны дополнительные функции маршрутизатора, в частности, функция трансляции адресов (NAT). Режим `router` может использоваться для подключения компьютеров локальной сети к внешней сети при ограниченном диапазоне публичных адресов. В этом случае на маршрутизаторе должно быть задано преобразование публичного адреса маршрутизатора в адрес внутреннего интерфейса модуля VPNM. Пример схемы сетевых подключений и настройки работы в этом режиме приведен в разделе [Настройка при работе маршрутизатора в режиме `router`](#) (на стр. 231).

Установка режима работы маршрутизатора и его настройка производится с помощью команд. Последовательность необходимых команд приведена при описании настройки каждого из режимов. После команд дана правильная конфигурация маршрутизатора для проверки произведенных настроек.

Команды приводятся вместе с приглашением для ввода команд. Приглашение содержит имя маршрутизатора, заключенное в угловые или квадратные скобки (например, `<USG2210>` или `[USG2210]`). Угловые скобки означают режим просмотра настроек, квадратные скобки – режим администрирования, позволяющий изменять настройки. Сами команды выделены жирным шрифтом.

По умолчанию имя маршрутизатора совпадает с названием его модели и может быть изменено с помощью соответствующей команды. Приведенные настройки гарантированно верны для модели Secoway USG2210, поэтому в описании команд присутствует имя USG2210. В случае изменения имени маршрутизатора в приглашении

для ввода команд будет присутствовать другое имя. За более подробными сведениями о синтаксисе команд и их назначении обратитесь к документации по маршрутизатору Huawei Secoway USG.

Настройка при работе маршрутизатора в режиме transparent

На приведенной ниже схеме представлен пример использования ПАК ViPNet Coordinator HW-VPNМ при работе маршрутизатора Huawei Secoway USG в режиме transparent. В примере предполагается, что:

- ПАК работает в автономном режиме (без использования внешнего межсетевого экрана);
- на одном из интерфейсов модуля VPNМ установлен публичный IP-адрес;
- к интерфейсу модуля VPNМ с публичным адресом (на схеме - eth1 (Ethernet1)) подключена внешняя сеть;
- к интерфейсу GigabitEthernet0/0/0 (на схеме - GE0/0/0) маршрутизатора Huawei Secoway USG подключена локальная сеть;
- второй интерфейс модуля VPNМ (на схеме - eth2 (Ethernet2)) может быть использован для подключения демилитаризованной зоны (на схеме - DMZ).

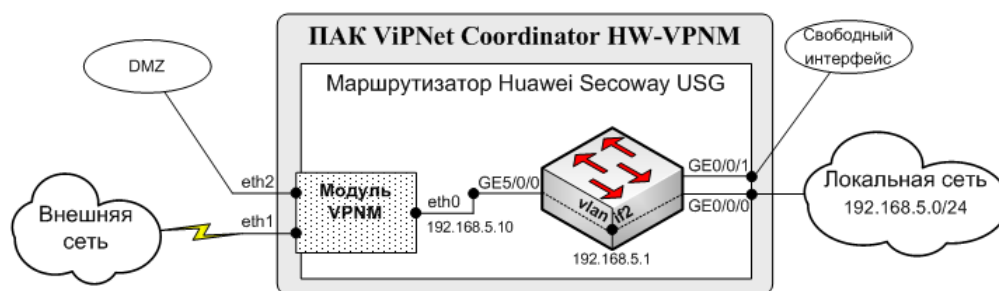


Рисунок 38: Схема подключения при работе маршрутизатора в режиме transparent

В режиме transparent маршрутизатор Huawei Secoway USG функционирует как сетевой коммутатор (свич), прозрачно пропуская трафик от узлов локальной сети до модуля VPNМ. Для обеспечения пропуска трафика интерфейсы маршрутизатора GigabitEthernet0/0/0 и GigabitEthernet5/0/0 объединяются в виртуальную локальную сеть, для которой создается виртуальный интерфейс (на схеме - интерфейс vlanif2). Адреса

виртуального интерфейса `vlanif2` и интерфейса `eth0` модуля `VPNM` должны принадлежать адресному пространству локальной сети.

На схеме используются следующие примеры адресов:

- локальная сеть имеет диапазон адресов `192.168.5.0/24`;
- интерфейс `vlanif2` имеет адрес `192.168.5.1`;
- интерфейс `eth0` имеет адрес `192.168.5.10`.

Для настройки ПО `ViPNet` в режиме работы без использования внешнего межсетевого экрана задайте в файле `iplir.conf` следующие параметры:

- В секции `[id]`, описывающей ПАК, параметр `usefirewall` установите в значение `off` (внешний межсетевой экран не используется).
- В секции `[dynamic]` параметр `dynamic_proxy` установите в значение `off`.
- Для всех используемых сетевых интерфейсов в секциях `[adapter]` установите параметр `type` в значение `internal`.

Конфигурирование маршрутизатора Huawei Secoway USG для работы в режиме `transparent` состоит из 3-х логических шагов, выполняемых в следующей последовательности:

1 Установка и конфигурирование режима `transparent`

Команда	Описание
<code><USG2210> system-view</code>	Вход в режим администрирования
<code>[USG2210] firewall mode transparent</code>	Установка режима <code>transparent</code>
You must delete config_file (vrpcfg.xxx) and reboot system!	Требуется перезагрузка системы
Are you sure?[Y/N] y	Подтвердите перезагрузку (нажмите Y)
<code>[USG2210] quit</code>	Выход из режима администрирования
<code><USG2210> reboot</code>	Перезагрузка системы
<code><USG2210> display firewall mode firewall mode transparent</code>	Просмотр текущего режима работы Убедитесь, что установлен нужный режим и

Команда	Описание
firewall mode transparent if reboot	что этот же режим будет устанавливаться после перезагрузки маршрутизатора
<USG2210> system-view	Вход в режим администрирования
[USG2210] firewall zone trust	Вход в режим конфигурирования зоны доверия
[USG2210-zone-trust] add interface GigabitEthernet0/0/0	Добавление интерфейса GigabitEthernet0/0/0 в зону доверия
[USG2210-zone-trust] quit	Выход из режима конфигурирования зоны доверия
[USG2210] firewall zone untrust	Вход в режим конфигурирования зоны недоверия
[USG2210-zone-untrust] add interface GigabitEthernet5/0/0	Добавление интерфейса GigabitEthernet5/0/0 в зону недоверия
[USG2210-zone-untrust] quit	Выход из режима конфигурирования зоны недоверия
[USG2210] firewall packet-filter default permit interzone trust untrust	Задание разрешающего правила фильтрации между зонами доверия и недоверия

2 Конфигурирование виртуальной локальной сети (VLAN)

Команда	Описание
[USG2210] vlan 2	Создание виртуальной локальной сети VLAN 2
[USG2210-vlan-2] quit	Возврат в режим администрирования
[USG2210] interface GigabitEthernet0/0/0	Вход в режим конфигурирования интерфейса GigabitEthernet0/0/0
[USG2210-GigabitEthernet0/0/0] port link-type access	Установка порта интерфейса GigabitEthernet0/0/0 в режим access
[USG2210-GigabitEthernet0/0/0] port access vlan 2	Включение порта интерфейса GigabitEthernet0/0/0 в виртуальную сеть VLAN 2
[USG2210-GigabitEthernet0/0/0] quit	Выход из режима конфигурирования интерфейса GigabitEthernet0/0/0
[USG2210] interface GigabitEthernet5/0/0	Вход в режим конфигурирования интерфейса GigabitEthernet5/0/0
[USG2210-GigabitEthernet5/0/0] port link-type access	Установка порта интерфейса GigabitEthernet5/0/0 в режим access
[USG2210-GigabitEthernet5/0/0] port access vlan 2	Включение порта интерфейса GigabitEthernet5/0/0 в виртуальную сеть VLAN 2

Команда	Описание
[USG2210-GigabitEthernet5/0/0] quit	Выход из режима конфигурирования интерфейса GigabitEthernet5/0/0

3 Конфигурирование интерфейса для виртуальной локальной сети

Команда	Описание
[USG2210] interface vlanif 2	Создание виртуального интерфейса для VLAN 2
[USG2210-vlanif2] ip address 192.168.5.1 255.255.255.0	Установка IP-адреса виртуального интерфейса (сервисный адрес, который используется для доступа к маршрутизатору)
[USG2210-vlanif2] quit	Выход из режима конфигурирования виртуального интерфейса
[USG2210] quit	Выход из режима администрирования

После настройки проверьте конфигурацию маршрутизатора с помощью команды `display current-configuration`. При правильной настройке конфигурация должна выглядеть следующим образом (в примере приведены только те данные, которые затрагивает произведенное конфигурирование):

```

#*****BEGIN*****public*****#
#
sysname USG2210
#
info-center timestamp debugging date
#
firewall packet-filter default permit interzone trust untrust direction inbound
firewall packet-filter default permit interzone trust untrust direction outbound
#
firewall blacklist filter-type icmp
firewall blacklist filter-type tcp
firewall blacklist filter-type udp
firewall blacklist filter-type others
#

```

```
firewall statistic system enable
#
vlan 2
#
interface Vlanif2
 ip address 192.168.5.1 255.255.255.0
#
interface GigabitEthernet0/0/0
 port link-type access
 port access vlan 2
#
interface GigabitEthernet0/0/1
 port link-type access
#
interface GigabitEthernet5/0/0
 port link-type access
 port access vlan 2
#
interface NULL0
#
firewall zone local
 set priority 100
#
firewall zone trust
 set priority 85
 add interface GigabitEthernet0/0/0
#
firewall zone untrust
 set priority 5
 add interface GigabitEthernet5/0/0
#
```

```
firewall zone dmz
set priority 50
#
return
#-----END-----#
```

Настройка при работе маршрутизатора в режиме router

На приведенной ниже схеме представлен пример использования ПАК ViPNet Coordinator HW-VPNМ при работе маршрутизатора Huawei Secoway USG в режиме `router`. В примере предполагается, что:

- ПАК работает в режиме со статической трансляцией адресов;
- на одном из интерфейсов маршрутизатора Huawei Secoway USG установлен публичный IP-адрес;
- к интерфейсу GigabitEthernet0/0/0 (на схеме - GE0/0/0) маршрутизатора Huawei Secoway USG подключена внешняя сеть;
- к одному из интерфейсов модуля VPNМ (на схеме - eth1 (Ethernet1)) подключена локальная сеть;
- второй интерфейс модуля VPNМ (на схеме - eth2 (Ethernet2)) может быть использован для подключения демилитаризованной зоны (на схеме - DMZ).

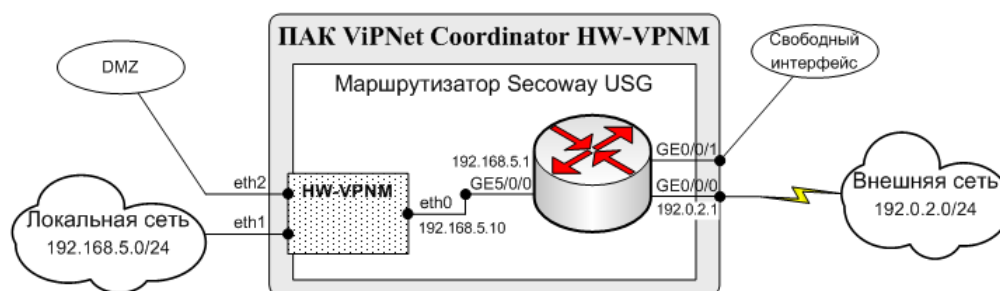


Рисунок 39: Схема подключения при работе маршрутизатора в режиме `router`

Адреса интерфейсов GigabitEthernet5/0/0 и eth0 должны принадлежать одному пространству частных адресов, при этом адрес интерфейса GigabitEthernet5/0/0 должен быть задан в качестве маршрута по умолчанию для интерфейса eth0. Адрес интерфейса GigabitEthernet0/0/0 должен принадлежать адресному пространству внешней сети. На маршрутизаторе должно быть задано правило трансляции адреса во внешней сети в адрес интерфейса eth0.

На схеме используются следующие примеры адресов:

- локальная сеть имеет диапазон адресов 192.168.5.0/24;
- интерфейс GigabitEthernet5/0/0 имеет адрес 192.168.5.1;
- интерфейс eth0 имеет адрес 192.168.5.10;
- внешняя сеть имеет диапазон адресов 192.0.2.0/24;
- интерфейс GigabitEthernet0/0/0 имеет адрес 192.0.2.1.

Для настройки ПО ViPNet в режиме работы со статической трансляцией адресов задайте в файле `iplir.conf` следующие параметры:

- В секции `[id]`, описывающей ПАК:
 - параметр `usefirewall` установите в значение `on` (используется внешний межсетевой экран);
 - параметр `proxyid` установите в значение `0x00000000`;
 - параметр `fixfirewall` установите в значение `off`.
- В секции `[dynamic]` параметр `dynamic_proxy` установите в значение `off`.
- В секции `[adapter]`, соответствующей сетевому интерфейсу `eth0`, параметр `type` установите в значение `external`.

Конфигурирование маршрутизатора Huawei Secoway USG для работы в режиме `router` осуществляется следующим образом:

Команда	Описание
<code><USG2210> system-view</code>	Вход в режим администрирования
<code>[USG2210] firewall mode router</code>	Установка режима <code>router</code>
You must delete config_file (vrpcfg.xxx) and reboot system!	Требуется перезагрузка системы
Are you sure?[Y/N] y	Подтвердите перезагрузку (нажмите Y)
<code>[USG2210] quit</code>	Выход из режима администрирования
<code><USG2210> reboot</code>	Перезагрузка системы
<code><USG2210> display firewall mode</code>	Просмотр текущего режима работы

Команда	Описание
firewall mode router	Убедитесь, что установлен нужный режим и
firewall mode router if reboot	что этот же режим будет устанавливаться после перезагрузки маршрутизатора
<USG2210> system-view	Вход в режим администрирования
[USG2210] firewall zone trust	Вход в режим конфигурирования зоны доверия
[USG2210-zone-trust] add interface GigabitEthernet5/0/0	Добавление интерфейса GigabitEthernet5/0/0 в зону доверия
[USG2210-zone-trust] quit	Выход из режима конфигурирования зоны доверия
[USG2210] interface GigabitEthernet5/0/0	Вход в режим конфигурирования интерфейса GigabitEthernet5/0/0
[USG2210- GigabitEthernet5/0/0] ip address 192.168.5.1 255.255.255.0	Установка IP-адреса интерфейса GigabitEthernet5/0/0
[USG2210- GigabitEthernet5/0/0] quit	Выход из режима конфигурирования интерфейса GigabitEthernet5/0/0
[USG2210] firewall zone untrust	Вход в режим конфигурирования зоны недоверия
[USG2210-zone-untrust] add interface GigabitEthernet0/0/0	Добавление интерфейса GigabitEthernet0/0/0 в зону недоверия
[USG2210-zone-untrust] quit	Выход из режима конфигурирования зоны недоверия
[USG2210] interface GigabitEthernet0/0/0	Вход в режим конфигурирования интерфейса GigabitEthernet0/0/0
[USG2210- GigabitEthernet0/0/0] ip address 192.0.2.1 255.255.255.0	Установка IP-адреса интерфейса GigabitEthernet0/0/0
[USG2210-GigabitEthernet0/0/0] quit	Выход из режима конфигурирования интерфейса GigabitEthernet0/0/0
[USG2210] firewall packet-filter default permit all	Задание разрешающего правила фильтрации между всеми зонами
[USG2210] nat server global 192.0.2.10 inside 192.168.5.10	Задание правила трансляции адреса
[USG2210] quit	Выход из режима администрирования

После настройки проверьте конфигурацию маршрутизатора с помощью команды `display current-configuration`. При правильной настройке конфигурация должна выглядеть следующим образом (в примере приведены только те данные, которые затрагивает произведенное конфигурирование):

```
#####BEGIN#####public#####  
#  
sysname USG2210  
#  
info-center timestamp debugging date  
#  
firewall packet-filter default permit interzone local untrust direction inbound  
firewall packet-filter default permit interzone local untrust direction outbound  
firewall packet-filter default permit interzone dmz untrust direction inbound  
firewall packet-filter default permit interzone dmz untrust direction outbound  
firewall packet-filter default permit interzone trust untrust direction inbound  
firewall packet-filter default permit interzone trust untrust direction outbound  
#  
nat server global 192.0.2.10 inside 192.168.5.10  
#  
firewall blacklist filter-type icmp  
firewall blacklist filter-type tcp  
firewall blacklist filter-type udp  
firewall blacklist filter-type others  
#  
firewall statistic system enable  
#  
interface GigabitEthernet0/0/0  
ip address 192.0.2.1 255.255.255.0  
#  
interface GigabitEthernet0/0/1  
#  
interface GigabitEthernet5/0/0  
ip address 192.168.5.1 255.255.255.0  
#  
interface NULL0
```

```
#  
firewall zone local  
  set priority 100  
#  
firewall zone trust  
  set priority 85  
  add interface GigabitEthernet5/0/0  
#  
firewall zone untrust  
  set priority 5  
  add interface GigabitEthernet0/0/0  
#  
firewall zone dmz  
  set priority 50  
#  
return  
#-----END-----#
```



Примеры настройки туннелей с использованием ПАК

В данном приложении рассмотрены две распространенные схемы использования туннелей в ViPNet, и для каждой схемы приведены необходимые настройки.

Схема 1

Пусть имеется два офиса, соединенных через Интернет. В одном из офисов находится сервер, к которому обращаются компьютеры из другого офиса, и необходимо, чтобы весь обмен данными через Интернет производился с шифрованием трафика. При этом установка ПО ViPNet непосредственно на участвующие в информационном обмене компьютеры невозможна или по каким-либо причинам нежелательна. Чтобы решить эту задачу, в каждом из офисов устанавливается ПАК ViPNet Coordinator HW, к которому подключаются компьютеры (см. схему).

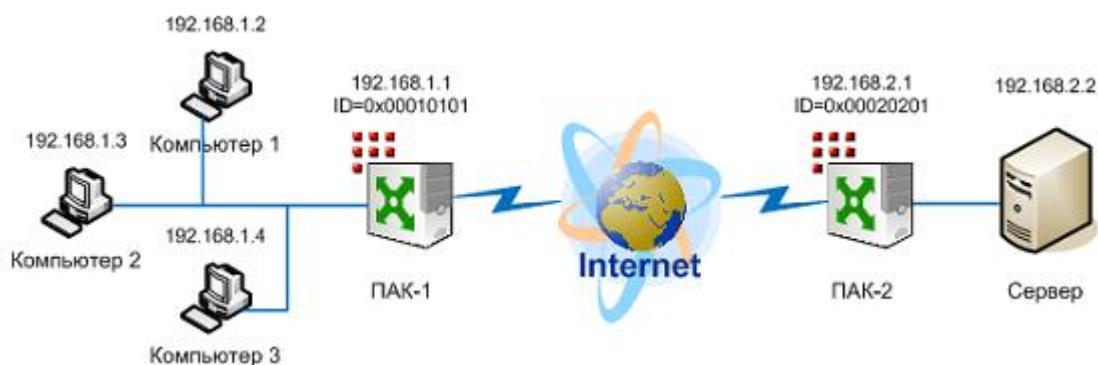


Рисунок 40: Схема настройки туннелей с использованием ПАК

Каждый из ПАК имеет два сетевых интерфейса, один из которых имеет реальный адрес и подключен к Интернету (внешний интерфейс), а второй имеет частный адрес и подключен к локальной сети офиса (внутренний интерфейс). Пусть ПАК-1 имеет внутренний адрес 192.168.1.1 и идентификатор в сети ViPNet 0x00010101, при этом подключенные к нему компьютеры 1, 2, 3 имеют адреса с 192.168.1.2 по 192.168.1.4. ПАК-2 имеет внутренний адрес 192.168.2.1 и идентификатор 0x00020201, а подключенный к нему сервер имеет адрес 192.168.2.2. Чтобы настроить правильную работу туннелей, на ПАК необходимо сделать следующие настройки в файле `iplir.conf` (команда `iplir config`):

- На ПАК-1:
 - В собственной секции `[id]` вписать строку:


```
tunnel= 192.168.1.2-192.168.1.4 to 192.168.1.2-192.168.1.4
```
 - В секции `[id]` для ПАК-2 вписать строку:


```
tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2
```
- На ПАК-2:
 - В собственной секции `[id]` вписать строку:


```
tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2
```
 - В секции `[id]` для ПАК-1 вписать строку:


```
tunnel= 192.168.1.2-192.168.1.4 to 192.168.1.2-192.168.1.4
```

Внутренние интерфейсы обоих ПАК не должны находиться в режиме 1. После запуска управляющего демона (команда `iplir start`) с указанными настройками компьютеры 1, 2, 3 смогут обращаться к серверу по адресу 192.168.2.2. При этом на участке между ПАК-1 и ПАК-2 пакеты будут идти в зашифрованном виде.

По умолчанию на обоих ПАК настроены правила, разрешающие трафик для всех туннелируемых адресов. Если в приведенном примере требуется закрыть доступ к серверу каким-либо компьютерам, необходимо изменить правила в файле `firewall.conf` и явно указать, каким компьютерам доступ разрешен. Например, пусть требуется закрыть доступ к серверу компьютеру 1 и разрешить доступ компьютерам 2 и 3. Для этого на ПАК-1 в секции `[tunnel]` файла `firewall.conf` необходимо удалить правило по умолчанию и вместо него задать следующие правила:

```
rule= proto any from 192.168.1.3-192.168.1.4 to 0x00020201 pass
rule= proto any from 0x00020201 to 192.168.1.3-192.168.1.4 pass
```

Эти правила разрешают трафик в обоих направлениях только между туннелируемыми компьютерами 2, 3 и ПАК-2, туннелирующим сервер. В свою очередь, на ПАК-2 должен быть разрешен трафик между сервером и ПАК-1. Для этого на ПАК-2 в секции `[tunnel]` файла `firewall.conf` можно оставить правило по умолчанию или вместо него задать правила, явно разрешающие трафик между туннелируемым сервером и ПАК-1 (в обоих направлениях):

```
rule= proto any from 192.168.2.2 to 0x00010101 pass
rule= proto any from 0x00010101 to 192.168.2.2 pass
```

В рассмотренном примере взаимодействие компьютеров с сервером разрешено по всем протоколам и портам. Чтобы ограничить это взаимодействие конкретными протоколами и портами, надо в приведенных правилах задать более жесткое условие. Например, пусть на сервере установлен веб-сервис, к которому разрешено обращаться компьютерам 2 и 3, но запрещено компьютеру 1. В этом случае на ПАК-1 в секции `[tunnel]` необходимо задать следующие правила:

```
rule= proto tcp from 192.168.1.3-192.168.1.4 to 0x00020201:80 pass
rule= proto tcp from 0x00020201:80 to 192.168.1.3-192.168.1.4 pass
```

На ПАК-2 в секции `[tunnel]` необходимо задать следующие правила:

```
rule= proto tcp from 192.168.2.2:80 to 0x00010101 pass
rule= proto tcp from 0x00010101 to 192.168.2.2:80 pass
```

Схема 2

Рассмотрим модифицированную схему использования туннелей, когда на компьютеры 1, 2, 3 установлено ПО `ViPNet Client`, а сервер остается незащищенным (см. схему). Пусть в сети `ViPNet` компьютерам 1, 2, 3 присвоены идентификаторы `0x00010102`, `0x00010103`, `0x00010104` соответственно.

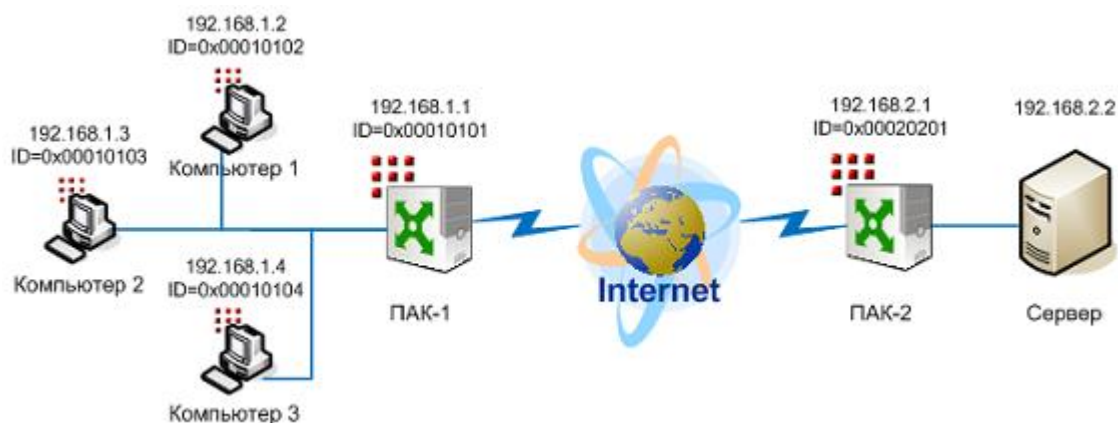


Рисунок 41: Модифицированная схема настройки туннелей

В этом случае, когда туннелируется только сервер, на ПАК необходимо сделать следующие настройки в файле `iplir.conf`:

- На ПАК-1:
 - В секции `[id]` для ПАК-2 вписать строку:


```
tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2
```
- На ПАК-2:
 - В собственной секции `[id]` вписать строку:


```
tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2
```

Если в настройках ПАК-2 задано правило по умолчанию для туннелируемых адресов, то никакие дополнительные настройки не нужны. Иначе на ПАК-2 в секции `[tunnel]` файла `firewall.conf` необходимо задать правила, разрешающие трафик между сервером и компьютерами 1, 2, 3:

```
rule= proto any from 192.168.2.2 to 0x00010102-0x00010104 pass
rule= proto any from 0x00010102-0x00010104 to 192.168.2.2 pass
```

Пусть требуется ограничить доступ к серверу со стороны некоторых узлов, например, закрыть доступ к серверу компьютеру 1 и разрешить доступ компьютерам 2 и 3. Для этого на ПАК-2 необходимо изменить правила в секции `[tunnel]` файла `firewall.conf` следующим образом:

```
rule= proto any from 192.168.2.2 to 0x00010103-0x00010104 pass
rule= proto any from 0x00010103-0x00010104 to 192.168.2.2 pass
```

Для случая, когда на сервере установлен веб-сервис, к которому разрешено обращаться компьютерам 2 и 3, но запрещено компьютеру 1, приведенные правила необходимо изменить следующим образом:

```
rule= proto tcp from 192.168.2.2:80 to 0x00010103-0x00010104 pass
```

```
rule= proto tcp from 0x00010103-0x00010104 to 192.168.2.2:80 pass
```


В

Примеры настроек работы ПАК через фиксированные альтернативные каналы

Рассмотрим упрощенную схему взаимодействия двух ПАК ViPNet Coordinator HW1000 (Координаторов), которые имеют два альтернативных канала доступа с условными названиями GlobalDataNet и Internet.

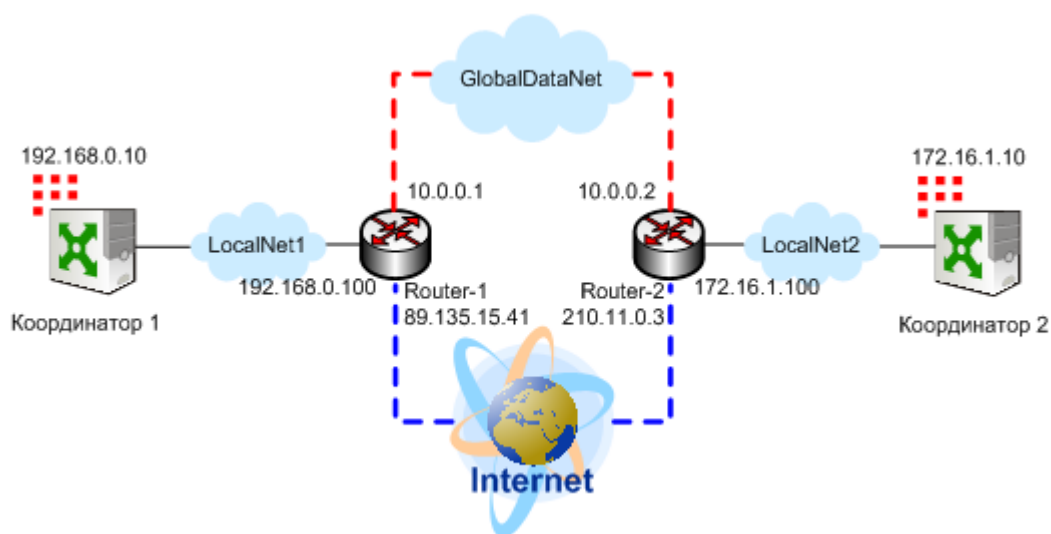


Рисунок 42: Схема взаимодействия двух ПАК ViPNet Coordinator HW1000 по двум каналам

Оба Координатора могут производить сетевой обмен без ограничений по любому из рассматриваемых каналов. Для работы через данные каналы подразумевается, что узлы Router-1 и Router-2 осуществляют маршрутизацию пакетов для данных каналов. При этом для пакетов, маршрутизируемых в Internet, будет выполняться подмена адреса отправителя, а для входящих из Internet пакетов с портом назначения 55777 будет выполняться подмена адреса получателя. На каждом Координаторе шлюз по умолчанию установлен на внутренний интерфейс соответствующего маршрутизатора. На обоих Координаторах установлен режим **Со статической трансляцией адресов** (см. «Настройка режима „Со статической трансляцией адресов“» на стр. 137).

Ниже приводятся несколько решений для типовых задач управления каналами взаимодействия между рассматриваемыми Координаторами.

1 Выбор канала GlobalDataNet в качестве фиксированного канала взаимодействия.

В первую очередь необходимо в файле `iplir.conf` для каждого из Координаторов создать секцию `[channels]`, в которой определить 2 альтернативных канала взаимодействия с привязкой к соответствующим группам:

```
[channels]
channel= GlobalDataNet, DataNetGroup
channel= Internet, InternetGroup
```

После этого необходимо задать привязку Координаторов к соответствующей группе и задать параметры доступа для каждого из каналов. Для этого в файле `iplir.conf` на Координаторе 1 в секции `[id]` для Координатора 2 необходимо задать следующие настройки:

```
group= DataNetGroup
channelfirewallip= GlobalDataNet, 10.0.0.2
channelfirewallip= Internet, 210.11.0.3
```

В файле `iplir.conf` на Координаторе 2 в секции `[id]` для Координатора 1 необходимо задать следующие настройки:

```
group= DataNetGroup
channelfirewallip= GlobalDataNet, 10.0.0.1
channelfirewallip= Internet, 89.135.15.41
```

После старта управляющего демона с данными настройками оба Координатора будут взаимодействовать только через канал GlobalDataNet.

2 Переключение на канал Internet.

Для переключения Координаторов на взаимодействие через канал Internet необходимо выполнить следующие настройки. В файле `iplir.conf` на Координаторе 1 в секции `[id]` для Координатора 2 необходимо изменить значение параметра `group` на следующее:

```
group= Internet
```

Аналогичные настройки необходимо произвести на Координаторе 2 в секции [id] для Координатора 1. После старта управляющего демона с данными настройками оба Координатора будут взаимодействовать только через канал Internet.

3 Отключение работы через фиксированные каналы.

Для отключения механизма работы через фиксированный канал в данном примере необходимо в файле `iplir.conf` на каждом Координаторе в секции [channels] удалить регистрацию групп в каналах, определенных параметрами `channel`:

```
[channels]
channel= GlobalDataNet
channel= Internet
```



Пример использования дополнительных IP-адресов на интерфейсе

В данном приложении приведен пример использования дополнительных IP-адресов на одном из интерфейсов ПАК ViPNet Coordinator HW.

Пусть в локальной сети находятся серверы, предоставляющие различные сервисы (например, почтовый сервер, веб-сервер и FTP-сервер). Требуется, чтобы эти сервисы были доступны из внешней сети, при этом серверы не имеют публичных адресов. Для решения этой задачи на границе локальной сети устанавливается ПАК ViPNet Coordinator HW, к которому подключаются серверы (см. схему).

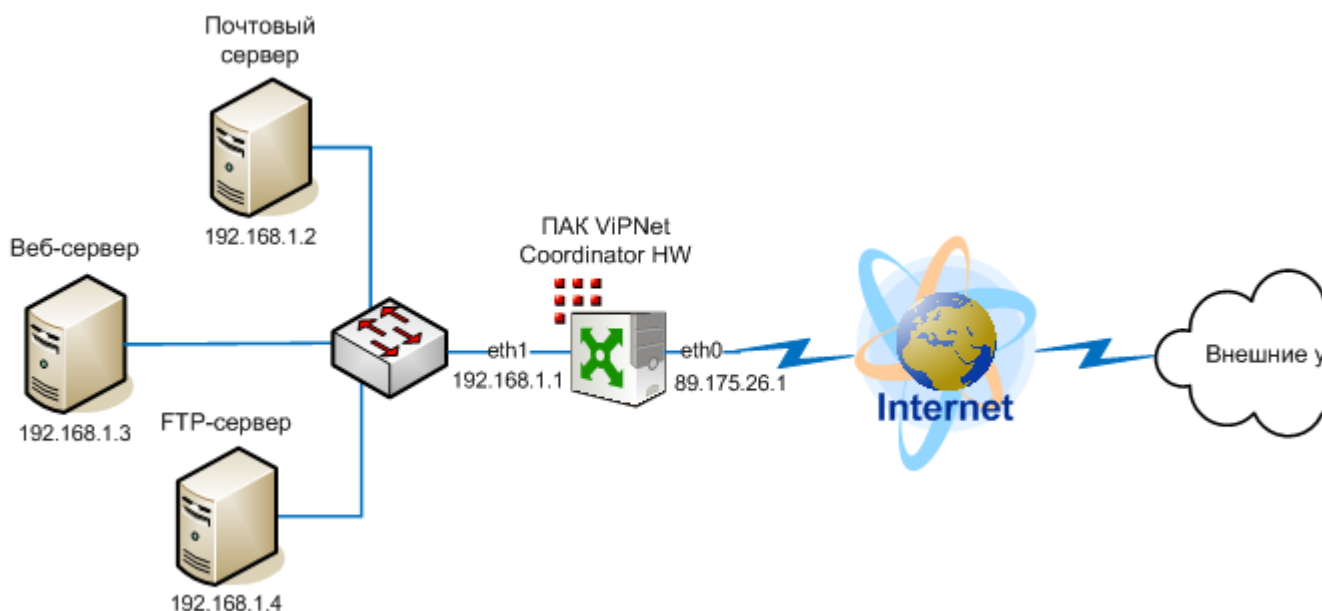


Рисунок 43: Схема подключения серверов к ПАК

ПАК имеет два сетевых интерфейса, один из которых имеет реальный адрес и подключен к Интернету (внешний интерфейс `eth0`), а второй имеет частный адрес и подключен к локальной сети (внутренний интерфейс `eth1`). Пусть ПАК имеет внешний адрес `89.175.26.1`, внутренний адрес `192.168.1.1`, а подключенные к нему серверы имеют частные адреса с `192.168.1.2` по `192.168.1.4`. Предполагается, что ПАК работает в режиме **Со статической трансляцией адресов**. Чтобы обеспечить доступ к серверам извне, на ПАК необходимо выполнить следующее:

- Задать дополнительные адреса на внешнем интерфейсе `eth0` с помощью следующих команд:

```
inet ifconfig eth0 address add 89.175.26.2 netmask 255.255.255.0
inet ifconfig eth0 address add 89.175.26.3 netmask 255.255.255.0
inet ifconfig eth0 address add 89.175.26.4 netmask 255.255.255.0
```

- В файле `firewall.conf` в секции `[nat]` задать следующие правила трансляции адреса получателя:

```
rule= change dst=192.168.1.2:25 proto any from anyip to 89.175.26.2:80
rule= change dst=192.168.1.3:80 proto any from anyip to 89.175.26.3:81
rule= change dst=192.168.1.4:21 proto any from anyip to 89.175.26.4:82
```

- В файле `firewall.conf` в секции `[forward]` задать следующие разрешающие правила:

```
rule= proto any from anyip to 192.168.1.2:25 pass
```

```
rule= proto any from anyip to 192.168.1.3:80 pass
rule= proto any from anyip to 192.168.1.4:21 pass
```

При указанных настройках к почтовому серверу можно будет обращаться по адресу 89.175.26.2 и номеру порта 80, к веб-серверу – по адресу 89.175.26.3 и номеру порта 81, к FTP-серверу – по адресу 89.175.26.4 и номеру порта 82. Приведенная ниже схема иллюстрирует организацию доступа к серверам с использованием дополнительных адресов на внешнем интерфейсе ПАК.

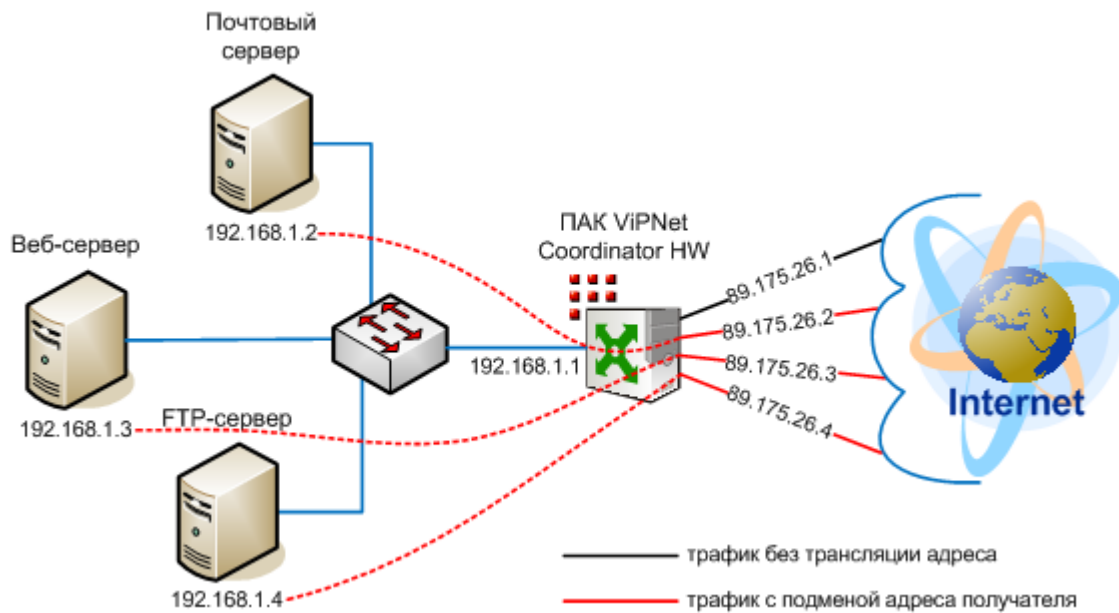


Рисунок 44: Схема организации доступа к серверам с помощью дополнительных адресов