ViPNet Coordinator Linux 4

Руководство администратора

infotecs

1991–2015 ОАО «ИнфоТеКС», Москва, Россия

ФРКЕ.00132-02 32 01

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 зарегистрированным товарным знаком ОАО «ИнфоТеКС».

В продукте использованы изобретения, защищенные патентами РФ №№ 2517411, 2526282, 2507569.

Все названия компаний и продуктов, которые являются товарными знаками или зарегистрированными товарными знаками, принадлежат соответствующим владельцам.

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

Сайт компании «ИнфоТеКС»: http://www.infotecs.ru

Электронный адрес службы поддержки: hotline@infotecs.ru

Содержание

Введение ... **8**
О документе ... 9
Для кого предназначен документ ... 9
Соглашения документа ... 9
О программе ... 11
Состав программного обеспечения ... 11
Системные требования ... 12
Комплект поставки ... 14
Новые возможности версии 4.1.4 ... 15
Обратная связь ... 16

Глава 1. Общие сведения ... **17**
Область применения ... 18
Функции координатора в защищенной сети ViPNet ... 19

Глава 2. Установка и обновление ViPNet Coordinator Linux ... **21**
Порядок установки ViPNet Coordinator Linux ... 22
Методы перехвата сетевых пакетов ... 23
Использование патча ядра ... 23
Использование технологии NETFILTER ... 25
Установка ПО ViPNet Coordinator Linux ... 27
Ручное конфигурирование ViPNet Coordinator Linux ... 30
Пересборка ядра Linux ... 32
Обновление версии ПО ViPNet Coordinator Linux ... 33
Конвертация правил открытой сети при обновлении до версии 4.x ... 34
Конвертация правил защищенной сети при обновлении до версии 4.x ... 37
Переход от использования фиксированных альтернативных каналов к множественным адресам доступа ... 39
Удаление ViPNet Coordinator Linux ... 41

Глава 3. Работа с ViPNet Coordinator Linux ... **42**
Управление ViPNet Coordinator Linux ... 43
Командный интерпретатор ViPNet Coordinator Linux ... 44
Обновление справочников и ключей ... 46

Глава 4. Настройка параметров защищенного подключения ... **47**

Общие принципы настройки 48
Настройка параметров защищенной сети 49
Секция [id] 49
Секция [adapter] 55
Секция [dynamic] 57
Секция [misc] 58
Секция [servers] 61
Секция [virtualip] 61
Секция [visibility] 62
Правила определения видимости сетевых узлов 63
Принципы назначения виртуальных адресов 64
Секция [debug] 65
Настройка параметров сетевых интерфейсов 66
Настройка режимов работы через межсетевой экран 68
Настройка режима «Без использования межсетевого экрана» 68
Настройка режима «Координатор» 68
Настройка режима «Со статической трансляцией адресов» 70
Настройка режима «С динамической трансляцией адресов» 71
Настройка координатора, выполняющего функции сервера открытого Интернета 74

Глава 5. Настройка параметров фильтрации IP-трафика 77
Основные принципы фильтрации трафика 78
Общие сведения о сетевых фильтрах 81
Работа с политиками безопасности 84
Использование групп объектов 85
Системные группы объектов 86
Пользовательские группы объектов, настроенные по умолчанию 87
Создание групп объектов 90
Просмотр групп объектов 91
Удаление групп объектов 92
Компоненты сетевых фильтров 93
Условие 93
Расписание 96
Действие 97
Ограничения на условия сетевых фильтров и правил трансляции 97
Создание сетевого фильтра 99
Просмотр сетевых фильтров 101
Изменение сетевого фильтра 104
Удаление сетевого фильтра 105

Антиспуфинг ... 107
Дополнительные опции межсетевого экрана ... 109

Глава 6. Настройка трансляции сетевых адресов (NAT) ... 110
Зачем используется трансляция адресов ... 111
Трансляция адресов в технологии ViPNet ... 112
Трансляция адреса назначения ... 112
Трансляция адреса источника ... 113
Компоненты правил трансляции адресов ... 115
Создание правила трансляции IP-адресов ... 116
Просмотр, изменение, удаление правил трансляции адресов ... 118

Глава 7. Настройка параметров обработки прикладных протоколов ... 119
Общие сведения о прикладных протоколах ... 120
Описание прикладных протоколов ... 122
Настройка параметров обработки прикладных протоколов ... 123

Глава 8. Система защиты от сбоев ... 126
Назначение системы защиты от сбоев ... 127
Состав системы защиты от сбоев и принципы ее работы ... 128
Управление системой защиты от сбоев ... 130
Настройка системы защиты от сбоев ... 131

Глава 9. Консольные утилиты ... 133
Программа просмотра журнала регистрации IP-пакетов ... 134
Программа просмотра информации о защищенном узле ... 140
Программа просмотра информации о состоянии системы защиты от сбоев ... 142
Программа смены пароля пользователя ... 144
Программа распаковки дистрибутива ключей ... 145
Программа работы с конфигурациями ViPNet ... 146

Глава 10. Тонкая настройка ПО ViPNet Coordinator Linux ... 149
Для чего нужна тонкая настройка ... 150
Изменение основных номеров устройств, используемых драйверами ... 151
Выбор режима шифрования ... 152
Фрагментирование шифрованных ViPNet-пакетов ... 153
Настройка максимального размера очереди пакетов в драйвере ... 154
Управление числом потоков в драйвере ... 155
Выбор метода подсчета контрольной суммы пакета ... 157

Настройка системного времени ОС Linux при использовании ПО ViPNet ... 158
Ручное переназначение виртуальных адресов узлов ... 159

Глава 11. Протоколирование событий, ведение и просмотр журналов ... 160
Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP ... 161
Контроль дампов аварийного завершения программы ViPNet Linux ... 164
Журналы устранения неполадок ПО ViPNet Coordinator Linux ... 166

Приложение A. Процессы-демоны, входящие в состав ПО ViPNet Linux ... 169

Приложение B. Модули ядра, входящие в состав ПО ViPNet Linux ... 172

Приложение C. Изменения, производимые в системе при установке и удалении ViPNet Coordinator Linux ... 173

Приложение D. Особенности работы ViPNet Coordinator Linux на различных аппаратных конфигурациях ... 177

Приложение E. События, отслеживаемые ПО ViPNet Coordinator Linux ... 179

Приложение F. Особенности работы координатора на границе локальной сети ... 186

Приложение G. Практические примеры настройки ПО ViPNet Coordinator Linux ... 188
Примеры настройки туннелей с использованием координаторов ViPNet ... 188
Пример использования дополнительных IP-адресов на интерфейсе ... 191

Приложение H. Справочник команд ... 194
Команды группы firewall ... 194
Команды группы iplir option ... 195
Команды группы alg ... 196

Приложение I. История версий ... 197
Что нового в версии 4.1.3 ... 197
Что нового в версии 4.1.2 ... 198
Что нового в версии 4.1 ... 198
Что нового в версии 4.0 ... 199
Что нового в версии 3.7.4 ... 201
Что нового в версии 3.7.3 ... 202
Что нового в версии 3.7.1 ... 203
Что нового в версии 3.7.0 ... 203

Что нового в версии 3.6.4 ... 204
Что нового в версии 3.6.3 ... 205
Что нового в версии 3.6.2 ... 205
Что нового в версии 3.6.1 ... 205
Что нового в версии 3.6.0 ... 205
Что нового в версии 3.5.2 ... 207
Что нового в версии 3.5.1 ... 207
Что нового в версии 3.5.0 ... 207
Что нового в версии 3.4.1 ... 208
Что нового в версии 3.4.0 ... 208

Приложение J. Глоссарий ... 210

Приложение K. Указатель ... 213

Введение

О документе 9

О программе 11

Новые возможности версии 4.1.3 15

Обратная связь 16

О документе

В данном документе описано назначение и применение ViPNet Coordinator Linux, возможности его установки и настройки, а также общие сведения о сетях ViPNet.

Для кого предназначен документ

Данный документ предназначен для администраторов, отвечающих за установку, настройку и эксплуатацию ПО ViPNet Coordinator Linux.

Предполагается, что администратор обладает знаниями и опытом в области сетевых технологий, достаточными для развертывания локальной сети, умеет устанавливать и настраивать операционные системы Linux, а также производить настройку межсетевых экранов.

Соглашения документа

Ниже перечислены соглашения, принятые в этом документе для выделения информации.

Таблица 1. Обозначения, используемые в примечаниях

Обозначение	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.

Таблица 2. Обозначения, используемые для выделения информации в тексте

Обозначение	Описание
Название	Название элемента интерфейса. Например, заголовок окна, название поля, кнопки или клавиши.
Клавиша+Клавиша	Сочетание клавиш. Чтобы использовать сочетание клавиш, следует нажать первую клавишу и, не отпуская ее, нажать вторую клавишу.
Меню > Подменю > Команда	Иерархическая последовательность элементов. Например, пункты меню или разделы на панели навигации.

Обозначение	Описание
`Код`	Имя файла, путь, фрагмент текстового файла (кода) или команда, выполняемая из командной строки.

При описании команд в данном документе используются следующие условные обозначения:

- Команды, которые могут быть выполнены только в режиме администратора, выделены красным цветом. Например:

  `команда`

- Параметры, которые должны быть заданы пользователем, заключены в угловые скобки. Например:

  `команда <параметр>`

- Необязательные параметры или ключевые слова заключены в квадратные скобки. Например:

  `команда <обязательный параметр> [необязательный параметр]`

- Если при вводе команды можно указать один из нескольких параметров, допустимые варианты заключены в фигурные скобки и разделены вертикальной чертой. Например:

  `команда {вариант-1 | вариант-2}`

О программе

Программное обеспечение ViPNet Coordinator Linux входит в состав комплекса продуктов ViPNet. ПО ViPNet Coordinator Linux устанавливается на компьютеры, которые выполняют функции серверов в защищенной сети ViPNet (см. «Функции координатора в защищенной сети ViPNet» на стр. 19).

Состав программного обеспечения

В состав ПО ViPNet Coordinator Linux 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 передачу необходимых параметров драйверу iplir, рассылку и прием информации об IP-адресах клиентов, ведение журнала трафика и тому подобное (см. «Настройка параметров защищенного подключения» на стр. 47). 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. 128). Драйвер работает на низком уровне и следит за работоспособностью компонентов ПО ViPNet Coordinator Linux.
- **Демон failoverd**, который обеспечивает функциональность системы защиты от сбоев (см. «Система защиты от сбоев» на стр. 126).
- **Демон mftpd (транспортный модуль MFTP)**, который обеспечивает прием и передачу транспортных конвертов между узлами сети ViPNet.
- **Демон algd**, который осуществляет обработку прикладных протоколов (см. «Настройка параметров обработки прикладных протоколов» на стр. 119).
- **SNMP-демон**, который позволяет получать статистику работы ПО ViPNet с удаленных узлов по протоколу SNMP (см. «Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP» на стр. 161).
- **Консольные утилиты**, позволяющие просматривать информацию об объекте сети ViPNet, журнал IP-трафика, работать с конфигурациями настроек, просматривать информацию о состоянии системы защиты от сбоев, изменять пароль пользователя ПО ViPNet Coordinator Linux и распаковывать дистрибутивы ключей.
- **Демон axis2.cgi**, обеспечивающий функциональность сервера веб-интерфейса.

Системные требования

Для установки ПО ViPNet Coordinator Linux необходим компьютер с характеристиками, описанными в разделах ниже.

Требования к компьютеру с архитектурой процессора x86

- Процессор — Intel Core Duo или другой схожий по производительности x86-совместимый процессор.
- Объем оперативной памяти — не менее 1 Гбайт.
- Свободное место на жестком диске — не менее 300 Мбайт.
- На компьютере должна быть установлена ОС Linux одного из следующих дистрибутивов:
  - ALT Linux 6.0 Server;
  - ALT Linux 6.0 Desktop;
  - ALT Linux СПТ 7.0;
  - Astra Linux Special Edition 1.3;
  - Astra Linux Special Edition 1.4;
  - CentOS 5.7 (32/64-разрядная);
  - CentOS 6.4 (32/64-разрядная);
  - Mandriva Linux 2010 Powerpack;
  - RedHat Enterprise Linux 5.7 (32/64-разрядная);
  - RedHat Enterprise Linux 6.4 AS (32/64-разрядная);
  - Slackware Linux 12.0 (только ядро 2.6.16.52 с FTP-сервера ftp://kernel.org);
  - Slackware Linux 12.2;
  - SUSE Linux Enterprise Server 10 SP4 (32/64-разрядная);
  - SUSE Linux Enterprise Server 11 SP3 (32/64-разрядная);
  - Ubuntu 12.04 (32/64-разрядная).

Требования к компьютеру с архитектурой процессора ARM

- Процессор — ARM 7.
- Объем оперативной памяти — не менее 1 Гбайт.
- Свободное место на жестком диске — не менее 300 Мбайт.
- На компьютере должна быть установлена ОС Linux одного из следующих дистрибутивов:
  - Debian 7 wheezy;
  - Ubuntu 12.04;

- Picuntu 12.04.

Дополнительные требования

Работа ПО на других дистрибутивах Linux возможна, но не гарантируется. Также не гарантируется нормальная работа ПО ViPNet Coordinator Linux на одном компьютере совместно со сторонними средствами преобразования трафика, со сторонними средствами защиты, работающими на канальном, сетевом или транспортном уровне (в частности, ipchains и iptables), и со сторонними средствами шифрования трафика (IPSec и так далее).

В системе должно использоваться ядро Linux из подмножества версий 2.6.x (от 2.6.18 до 2.6.39 включительно) или 3.x (до 3.11 включительно).

Примечание. При использовании патча ядра в качестве метода перехвата сетевых пакетов драйвером ViPNet (см. «Использование патча ядра» на стр. 23) не поддерживается совместимость с исходными текстами ядер Linux, которые поставляются в составе дистрибутивов. В этом случае гарантируется совместимость только с официальными версиями ядер Linux с FTP-сервера ftp://kernel.org.

Размер временного файлового хранилища tmpfs, если оно смонтировано на `/tmp`, должен быть не меньше 2 Гбайт. Размер tmpfs настраивается по-разному в зависимости от дистрибутива ОС Linux, подробнее об этом см. в man-страницах операционной системы.

Для работы ПО необходимы следующие установленные пакеты:

Название пакета	Версия, не ниже	Версия, не выше
Ядро Linux	2.6.18	3.11
awk – утилита для обработки текстовых данных	3.1.5	
bash – командный интерпретатор	1.10	
cron – системный планировщик задач, необходим автоматический запуск при старте системы для работы функционала ротации логов (см. «Журналы устранения неполадок ПО ViPNet Coordinator Linux» на стр. 166)	3.0	
ethtool – утилита конфигурирования параметров сетевых интерфейсов	1.3	
fileutils – пакет для работы с файлами	4.0	
grep – программа анализа строк	2.3	
logrotate – утилита ротации логов	3.0	
modutils – пакет для работы с модулями	2.1	
net-tools – программы работы с сетью	1.53	

Название пакета	Версия, не ниже	Версия, не выше
psmisc – программы работы с процессами	18.3	
sed – потоковый текстовый редактор	4.0.9	
sh-utils – пакет для командного интерпретатора	2.0	
sysklogd – менеджер системных логов	1.3	

Для установки ПО дополнительно к указанным необходимы следующие пакеты (их можно удалить после успешной установки):

Название пакета	Версия, не ниже	Версия, не выше
Заголовочные файлы ядра Linux (исходные тексты в случае использования технологии patch для перехвата сетевых пакетов)	2.6.18	3.11
gcc – компилятор Си	Серии 4.x не ниже 4.0.2	
gzip – компрессор файлов	1.2	
make – обработчик скриптов сборки	3.75	
patch – программа установки патча	2.5	
tar – архиватор	1.13	

Комплект поставки

В комплект поставки ПО ViPNet Coordinator Linux входят:

- Дистрибутив ПО ViPNet Coordinator Linux.
- Документ «ViPNet Coordinator Linux. Руководство администратора» в формате PDF.

Новые возможности версии 4.1.4

В ViPNet Coordinator Linux версии 4.1.4 исправлены ошибки, выявленные при эксплуатации версии 4.1.3. Информация об изменениях в предыдущих версиях программы приведена в приложении История версий (на стр. 197).

Обратная связь

Дополнительная информация

Сведения о продуктах и решениях ViPNet, распространенные вопросы и другая полезная информация собраны на сайте ОАО «ИнфоТеКС»:

- Веб-портал документации ViPNet http://docs.infotecs.ru.
- Описание продуктов ViPNet http://www.infotecs.ru/products/line/.
- Информация о решениях ViPNet http://www.infotecs.ru/solutions/.
- Сборник часто задаваемых вопросов (FAQ) http://www.infotecs.ru/support/faq/.
- Форум пользователей продуктов ViPNet http://www.infotecs.ru/forum.
- Законодательная база в сфере защиты информации http://www.infotecs.ru/laws/.

Контактная информация

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

- Техническая поддержка для пользователей продуктов ViPNet: hotline@infotecs.ru.
- Форма запроса в службу технической поддержки http://www.infotecs.ru/support/request/.
- Регистрация продуктов и консультации по телефону для клиентов, имеющих расширенный уровень технического сопровождения:

  8 (495) 737-6196,

  8 (800) 250-0260 — бесплатный звонок из любого региона России (кроме Москвы).

Распространение информации об уязвимостях продуктов ОАО «ИнфоТеКС» регулируется политикой ответственного разглашения http://infotecs.ru/products/disclosure.php. Если вы обнаружили уязвимости в продуктах компании, сообщите о них по адресу security-notifications@infotecs.ru.

1

Общие сведения

Область применения 18

Функции координатора в защищенной сети ViPNet 19

Область применения

Комплекс программных продуктов ViPNet является универсальным инструментом для развертывания виртуальных защищенных сетей (VPN) любых конфигураций, обеспечивающих прозрачное взаимодействие компьютеров, включенных в сеть ViPNet, независимо от способа, места и типа выделяемого адреса при их подключении к сети.

Информация, которой обмениваются участники сети, недоступна для других пользователей, не участвующих в обмене. Данные, хранящиеся на компьютерах ViPNet-сети, надежно защищены от несанкционированного доступа как с компьютеров корпоративной сети, так и с компьютеров, не входящих в сеть ViPNet.

Виртуальная сеть ViPNet (см. «Виртуальная защищенная сеть» на стр. 210) строится путем установки на компьютеры (сетевые узлы) ПО ViPNet Client (рабочие места пользователей — клиенты) и ПО ViPNet Coordinator (серверы защищенной сети — координаторы). ViPNet Client обеспечивает сетевую защиту и включение в VPN-сеть отдельных компьютеров. Компьютер с ПО ViPNet Coordinator (Linux или Windows) обычно устанавливается на границах локальных сетей и их сегментов и обеспечивает:

- включение в корпоративную сеть открытых и защищенных компьютеров, находящихся в этих локальных сетях, независимо от способа подключения и типа IP-адреса компьютера;
- разделение и защиту сетей от сетевых атак;
- оповещение клиентских компьютеров о состоянии других сетевых узлов, связанных с ним.

Кроме координаторов и клиентов, в состав защищенной сети ViPNet входят административные центры — сетевые узлы с установленным ПО ViPNet Administrator и ViPNet Policy Manager. Узел, на котором установлено ПО ViPNet Administrator, выполняет функции Центра управления сетью и Удостоверяющего и ключевого центра. Эти административные центры отвечают за конфигурирование сети, возможность организации защищенных связей между объектами сети, генерацию ключей, используемых для шифрования, и так далее. На узел, выполняющий функции Центра управления сетью, может быть установлена программа ViPNet Policy Manager (на стр. 210), которая позволяет формировать корпоративную политику безопасности и рассылать ее на сетевые узлы.

ПО ViPNet Coordinator Linux работает в сетях, построенных на базе следующих протоколов канального уровня:

- Ethernet;
- PPP;
- SLIP (включая CSLIP, SLIP6, CSLIP6);
- Wireless-протоколов стандарта IEEE 802.11 (Wi-Fi).

Функции координатора в защищенной сети ViPNet

VPN-сервер в защищенной сети ViPNet называется координатором. В зависимости от задач, которые требуется решить в рамках сети ViPNet, координатор может выполнять следующие функции:

- **Сервер IP-адресов** — функция, которая в автоматическом режиме обеспечивает взаимодействие защищенных узлов (клиентов и координаторов) как внутри данной виртуальной сети, так и при взаимодействии с другими виртуальными сетями ViPNet. Это возможно благодаря использованию специального протокола динамической маршрутизации VPN-трафика, реализующего обмен информацией о параметрах доступа узлов друг к другу. Данный протокол обеспечивает маршрутизацию VPN-трафика между узлами в сети ViPNet методом, наиболее оптимальным для используемого способа подключения узла к сети.
- **Маршрутизатор VPN-пакетов** — функция, обеспечивающая маршрутизацию транзитного VPN-трафика, проходящего через координатор, на другие защищенные узлы. Маршрутизация осуществляется на основании идентификаторов защищенных узлов, содержащихся в открытой части VPN-пакетов, которая защищена от подделки, и на основании защищенного протокола динамической маршрутизации VPN-трафика. Одновременно выполняется функция трансляции адресов для VPN-трафика, и все пакеты, поступающие на координатор, отправляются на другие узлы от имени IP-адреса координатора.
- **VPN-шлюз** — стандартная для классических VPN функция, реализующая создание защищенных каналов (туннелей) посредством шифрования трафика открытых узлов, размещенных за координатором, и передачи этого трафика на другие VPN-шлюзы или защищенные клиенты. VPN-шлюз интегрирован с межсетевым экраном для защищенных и открытых соединений, который осуществляет фильтрацию незашифрованного трафика, а также трафика внутри защищенного соединения.
- **Сервер-маршрутизатор** — функция, которая обеспечивает доставку на сетевые узлы управляющих сообщений, обновлений ключей и программного обеспечения из программы ViPNet Центр управления сетью, а также обмен прикладными транспортными конвертами между узлами (см. «Транспортный конверт» на стр. 211).

  Маршрутизация прикладных и управляющих конвертов осуществляется с помощью транспортного модуля ViPNet MFTP, работающего на прикладном уровне. Транспортный модуль на координаторе принимает конверты от других узлов сети ViPNet и пересылает их на узел назначения.

  Маршрутизация данных между координаторами выполняется на основании межсерверных каналов, заданных для этих координаторов. Межсерверные каналы могут быть организованы по любой схеме. Если маршрутов несколько, передача информации осуществляется по кратчайшему из них. Передача информации из одной сети ViPNet в другую выполняется через шлюзовые координаторы, с помощью которых происходит взаимодействие двух сетей.

- **Межсетевой экран** — функция, благодаря которой координатор выполняет фильтрацию открытых транзитных и локальных сетевых соединений по IP-адресам, протоколам, портам, направлениям соединений и другим параметрам. Одновременно координатор может выполнять функции трансляции адресов для проходящего через него открытого трафика (см. «Трансляция сетевых адресов (NAT)» на стр. 211).

  Функция трансляции адресов для открытого трафика позволяет задать правила трансляции адресов для решения двух основных задач:

  - Подключение локальной сети к открытым ресурсам Интернета, когда количество узлов локальной сети превышает количество публичных IP-адресов, выданных поставщиком услуг Интернета.
  - Организация доступа к открытым серверам локальной сети из Интернета.

- **Сервер Открытого Интернета** — функция, которая позволяет обеспечить раздельный доступ защищенных узлов в Интернет и к ресурсам защищенной сети ViPNet, если этого требует политика безопасности организации. Защищенные узлы, которые имеют связь с сервером Открытого Интернета, могут работать в одном из двух режимов:

  - Доступ к защищенной сети ViPNet при отсутствии подключения к Интернету.
  - Доступ в Интернет при отсутствии соединения с защищенными узлами ViPNet.

2

Установка и обновление ViPNet Coordinator Linux

Порядок установки ViPNet Coordinator Linux 22

Методы перехвата сетевых пакетов 23

Установка ПО ViPNet Coordinator Linux 27

Ручное конфигурирование ViPNet Coordinator Linux 30

Обновление версии ПО ViPNet Coordinator Linux 33

Удаление ViPNet Coordinator Linux 41

Порядок установки ViPNet Coordinator Linux

Внимание! Все действия по установке ViPNet Coordinator Linux могут производиться только пользователем, имеющим права администратора (root).

Для установки ViPNet Coordinator Linux выполните действия, приведенные в таблице ниже.

Таблица 3. Порядок установки ViPNet Coordinator Linux

	Действие	Ссылка
☐	Проверьте наличие дополнительно устанавливаемых пакетов для работы с ПО ViPNet Coordinator Linux.	Системные требования (на стр. 12)
☐	Выберите метод перехвата сетевых пакетов, который вы будете использовать.	Методы перехвата сетевых пакетов (на стр. 23)
☐	Если вы выбрали метод перехвата пакета с использованием патча ядра, установите специальный патч ядра, поставляемый с дистрибутивом ViPNet Coordinator Linux, и пересоберите ядро Linux.	Использование патча ядра (на стр. 23)
☐	Установите ПО ViPNet Coordinator Linux.	Установка ПО ViPNet Coordinator Linux (на стр. 27)
☐	При необходимости измените конфигурацию ViPNet Coordinator Linux.	Ручное конфигурирование ViPNet Coordinator Linux (на стр. 30)

Совет. Мы рекомендуем распечатать список и отмечать в нем шаги по мере их выполнения.

Методы перехвата сетевых пакетов

ViPNet Coordinator Linux поддерживает два метода перехвата сетевых пакетов:

- Использование патча ядра (на стр. 23). Данный метод требует, чтобы перед установкой ПО был установлен специальный патч ядра, поставляемый с дистрибутивом ViPNet Coordinator Linux. После установки патча ядро Linux необходимо пересобрать, и только после этого производить установку ViPNet Coordinator Linux. Кроме того, при использовании данного метода не поддерживается совместимость с исходными текстами ядер Linux, которые поставляются в составе дистрибутивов.
- Использование технологии NETFILTER (на стр. 25). В случае использования технологии NETFILTER установка патча не требуется, функционал перехвата пакетов является стандартным компонентом любого из поддерживаемых ядер Linux. В большинстве случаев необходимые параметры ядра включены по умолчанию, и пересборка ядра не требуется. При выборе данного метода перехвата пакетов можно использовать ядра, поставляемые с поддерживаемыми дистрибутивами Linux, а не только официальные версии ядер.

Выбор метода встраивания ViPNet Coordinator Linux остается за пользователем. Если в процессе работы требуется изменить метод перехвата пакетов, то необходимо активировать нужный метод и переустановить ViPNet Coordinator Linux.

Внимание! В случае включения обоих описанных выше методов в процессе работы ViPNet Coordinator Linux будет использоваться патч ядра.

Использование патча ядра

При использовании данного метода перехвата сетевых пакетов не поддерживается совместимость с исходными текстами ядер Linux, которые поставляются в составе дистрибутивов. Создатели дистрибутивов вносят в них изменения, часто непродуманные, которые приводят к неправильной сборке и работе модулей ядра, собранных для работы на стандартном ядре. Компания «ИнфоТеКС» рекомендует всегда загружать только официальную версию ядра Linux с FTP-сервера ftp://kernel.org (последнее ядро нужной серии), и именно ее пересобирать под свои нужды и устанавливать на ней ПО ViPNet. Также необходимо помнить, что пересобирать ядро и устанавливать ViPNet Coordinator Linux нужно обязательно с использованием одной и той же версии компилятора.

Для установки патча ядра Linux выполните следующие действия:

1. Распакуйте дистрибутив ViPNet Coordinator Linux в любой каталог командой `tar xvfz <file>`. При распаковке будет создан каталог `distribute`, внутри которого в подкаталоге `patch` находятся патчи для разных версий ядра Linux.
2. Убедитесь, что в каталоге `/usr/src/linux` находятся исходные тексты для используемой версии ядра.

3 Скопируйте файл патча для используемой версии ядра Linux из каталога `distribute/patch` в каталог `/usr/src`. В настоящий момент поддерживаются ядра серии 2.6.x и 3.x. В зависимости от используемой версии ядра выберите соответствующий ей файл патча:

   - для версий 2.6.18 по 2.6.26 – файл `iplir.patch.2.6.13`;
   - для версий 2.6.27 и выше – файл `iplir.patch.2.6.27+`;
   - для версий 3.x — файл `iplir.patch.2.6.39+`.

4 Войдите в каталог `/usr/src/<каталог с ядром>` и установите патч командой `patch -p1 < файл_патча`. Если программа `patch` сообщает об ошибках при установке патча, это означает, что либо ядро Linux на компьютере не относится к поддерживаемым сериям, либо оно содержит какие-либо нестандартные модификации, либо неправильно выбран файл патча. В этом случае необходимо установить стандартное ядро Linux одной из поддерживаемых серий, выбрать правильный файл в соответствии с изложенным выше либо выбрать другой способ перехвата пакетов.

Если патч установился нормально, то необходимо войти в каталог `/usr/src/linux` и запустить конфигуратор ядра – командой `make menuconfig` для текстового режима или командой `make xconfig` для графического. В конфигураторе нужно установить следующие опции:

- В разделе **Networking options** необходимо включить опцию **IpLir Crypting network driver**.

  Для ядер версий 2.6.27 и выше эта опция называется **ViPNet low-level packet filter infrastructure**. Если опция включена, то будет доступна расположенная под ней опция **Block all packets when ViPNet drivers are not loaded**. Включение данной опции позволяет обеспечить безопасность компьютера при старте ОС, так как до загрузки ПО ViPNet Coordinator Linux все сетевые пакеты будут блокироваться.

- В разделе **Loadable module support** необходимо включить опцию **Loadable module support**.

  Рекомендуется включить также опцию **Set version information on all symbols in module**, что позволяет избежать потенциальных проблем при последующем обновлении ядра. Включение этой опции не является обязательным, однако не рекомендуется выключать ее, если вы не являетесь профессиональным администратором Linux и не очень хорошо представляете себе, зачем это нужно. Для ядер серии 2.6.x данная опция называется **Module versioning support**.

- Остальные опции устанавливаются по желанию администратора и в соответствии со здравым смыслом.

После выхода из конфигуратора нужно собрать ядро. Для этого выполните следующие команды в приведенной последовательности:

```
make
make install
make modules_install
```

Внимание! После установки ядра проверьте, чтобы файл `/boot/System.map<версия_ядра>` или, в случае его отсутствия, файл `/boot/System.map` соответствовал собранному ядру.

Убедитесь, что после перезагрузки будет загружено только что собранное ядро. Для этого произведите необходимые изменения в конфигурационных файлах загрузчика операционной системы. Перезагрузите систему. Проверить наличие патча в загруженном ядре можно командой:

```
cat /proc/kallsyms | grep set_packet_filter | wc -l
```

или, если файл `/proc/kallsyms` отсутствует, командой:

```
cat /boot/System.map-`uname -r` |grep set_packet_filter | wc -l
```

Если эта команда выдает 1 или большее число, все установлено правильно, если 0, то патч не был нормально установлен или загружено не то ядро, которое пересобиралось.

Использование технологии NETFILTER

При использовании данного метода перехвата сетевых пакетов можно использовать как ядра, поставляемые в комплекте с поддерживаемыми дистрибутивами, так и официальные версии ядер Linux.

В большинстве случаев при использовании ядер, поставляемых в комплекте дистрибутива Linux, параметры ядра, необходимые для использования NETFILTER, уже включены по умолчанию и пересборка ядра не требуется. Проверить функционирование NETFILTER можно командой:

```
cat /proc/kallsyms | grep nf_reinject | wc -l
```

или, если файл /proc/kallsyms отсутствует, командой:

```
cat /boot/System.map-`uname -r` |grep nf_reinject | wc -l
```

Если эта команда выдает 1 или большее число, то NETFILTER включен и пересборка ядра не требуется, если 0, то необходимо сконфигурировать ядро с поддержкой NETFILTER и пересобрать его.

Для включения поддержки NETFILTER в ядре необходимо установить исходные тексты используемого ядра. Рекомендуется использовать официальную версию ядра Linux с FTP-сервера ftp://kernel.org, так как сборка ядра из исходных текстов, поставляемых с дистрибутивом Linux, не всегда проходит корректно. Далее нужно зайти в каталог с установленными исходными текстами ядра и запустить конфигуратор ядра – командой `make menuconfig` для текстового режима или `make xconfig` для графического режима. В конфигураторе нужно установить следующие опции:

- В разделе **Networking options** необходимо включить опцию **Network packet filtering**.
- В разделе **Netfilter Configuration** необходимо выключить опцию **IP tables support**.

После выхода из конфигуратора нужно собрать ядро. Для этого выполните следующие команды в приведенной последовательности:

```
make
make install
make modules_install
```

Внимание! После установки ядра проверьте, чтобы файл `/boot/System.map<версия_ядра>` или, в случае его отсутствия, файл `/boot/System.map` соответствовал собранному ядру.

Убедитесь, что после перезагрузки будет загружено только что собранное ядро. Для этого произведите необходимые изменения в конфигурационных файлах загрузчика операционной системы. Перезагрузите систему. Проверьте функционирование NETFILTER приведенным выше способом.

Внимание! При использовании технологии NETFILTER с дистрибутивом SUSE Linux Enterprise Server 10 без пересборки ядра необходимо установить пакет `kernel-source-2.6.16.21-08`, требуемый для сборки драйверов ViPNet.

Примечание. При использовании технологии NETFILTER события 82 и 89 (см. «События, отслеживаемые ПО ViPNet Coordinator Linux» на стр. 179) не фиксируются в журнале IP-пакетов, так как операционная система автоматически блокирует соответствующие IP-пакеты.

Установка ПО ViPNet Coordinator Linux

Установка производится пользователем, имеющим права администратора (root, user ID = 0). При установке необходимо иметь прямой доступ к консоли компьютера, на котором устанавливается ПО ViPNet Coordinator Linux, поскольку до окончательной настройки необходимых параметров компьютер может оказаться недоступным по сети.

Для установки необходимо иметь:

- дистрибутив ПО ViPNet Coordinator Linux;
- дистрибутив ключей, которые будут использоваться на данном компьютере. Этот дистрибутив необходимо получить от администратора сети ViPNet. Кроме того, администратор должен сообщить пароль пользователя сетевого узла.

Внимание! Чтобы избежать потенциальных проблем при взаимодействии с ViPNet Administrator и ViPNet Policy Manager (которые работают под ОС Windows), необходимо соблюдать несложное правило: все имена каталогов, в которые распаковывается дистрибутив ViPNet Coordinator Linux, в которых хранятся справочники, ключи и так далее, должны содержать только латинские символы нижнего регистра, цифры, а также знаки дефис, подчеркивание и тому подобное. Не используйте заглавные буквы в этих именах.

Для установки ПО ViPNet Coordinator Linux выполните следующие действия:

1. Войдите в каталог `distribute`, где находится распакованный дистрибутив ПО ViPNet Coordinator Linux, и выполните команду `./install.sh`. Программа установки производит окончательную сборку драйверов, установку драйверов в системе, а также установку прикладных программ и конфигурирование ViPNet Coordinator Linux.

   Далее установка состоит из нескольких шагов, следуйте инструкциям программы установки.

2. Анализ наличия требуемых для установки пакетов (см. «Системные требования» на стр. 12).

   Если требуемый пакет не найден в системе, программа установки завершает работу с выводом соответствующего сообщения. В этом случае установите недостающие пакеты и запустите программу установки повторно.

3. Выбор дистрибутива ПО ViPNet Coordinator Linux.

   На данном шаге программа установки выполняет поиск доступных для установки дистрибутивов в текущем каталоге. В стандартном случае в текущем каталоге находится только один доступный дистрибутив `distribute.tar.gz`, который и будет предложен для установки. В специфических случаях может возникнуть необходимость выбора другого дистрибутива. В этом случае следует указать каталог для поиска дистрибутива, повторить поиск и выбрать нужный дистрибутив.

4 Распаковка выбранного дистрибутива.

После успешного выбора дистрибутива программа установки производит его распаковку во временный каталог и после завершения распаковки выводит лицензионное соглашение. Чтобы перейти к следующему шагу, ознакомьтесь с лицензионным соглашением и подтвердите свое согласие с его условиями.

5 Анализ текущей установленной конфигурации ViPNet.

На этом шаге производится поиск установленной конфигурации ViPNet. Если такая конфигурация найдена, то вам будет задан вопрос – хотите ли вы использовать текущую конфигурацию. В случае положительного ответа конфигурация сохраняется, в случае отрицательного — в процессе дальнейшей установки будет предложено выбрать новую конфигурацию.

6 Подготовка к установке.

На данном шаге выполняется ряд дополнительных проверок и других действий, необходимых для дальнейшей установки (определение метода перехвата пакетов, поддержка дистрибутива Linux, компилятора, проверка доступного места на жестком диске и так далее).

Переход к следующему шагу возможен лишь при успешном окончании подготовительного этапа.

7 Установка драйверов.

На этом шаге производится окончательная компиляция драйверов под текущее ядро Linux и их установка.

8 Установка приложений.

На данном шаге производится установка приложений, скриптов, файлов конфигурации, конфигурация стартовых скриптов.

9 Сохранение параметров автоматического запуска приложений.

В случае, если на 4-ом шаге была обнаружена текущая установленная конфигурация ViPNet и вы решили ее сохранить, то после успешного окончания 8-го шага вам будет предложено запустить службы ViPNet. На этом процесс установки завершается. Если вы отказались стартовать службы, то запустите их вручную или перезагрузите компьютер.

10 Выбор дистрибутива ключей.

На данном шаге программа установки выполняет поиск доступных дистрибутивов ключей (файлов `*.dst`) в текущем каталоге. При необходимости выберите другой каталог для поиска или завершите установку. В случае завершения установки на текущем шаге выберите дистрибутив ключей вручную и распакуйте его в выбранный каталог командой `/sbin/unmerge <file.dst> <каталог>`, а затем сконфигурируйте ViPNet Coordinator Linux (см. «Ручное конфигурирование ViPNet Coordinator Linux» на стр. 30).

11 Выбор каталога для установки конфигурации.

Если на 4-ом шаге была обнаружена текущая установленная конфигурация, перезапишите ее новой конфигурацией в том же каталоге. В случае отказа или если установленной конфигурации найдено не было, выберите каталог для установки ключей. Если не будет указан

иной каталог, ключи будут установлены в каталог `/etc/vipnet` по умолчанию. После выбора каталога производится автоматическая распаковка дистрибутива ключей в этот каталог.

Примечание. Если на разделе жесткого диска, на который распаковывается дистрибутив ключей, свободного места меньше, чем необходимо, выводится сообщение об ошибке и установка завершается.

---

12 Аутентификация пользователя.

Для успешного завершения установки введите правильный пароль для выбранной конфигурации ViPNet. После ввода пароля осуществляется его проверка и сохранение в зашифрованном виде в случае правильного пароля. Если проверка пароля не прошла, попробуйте снова ввести пароль. В случае отказа от ввода пароля установка завершается некорректно. В этом случае для нормальной работы ПО ViPNet Coordinator Linux выполните ручное конфигурирование (см. «Ручное конфигурирование ViPNet Coordinator Linux» на стр. 30).

13 После успешного завершения последнего шага установки запустите службы ViPNet. На этом процесс установки завершается. В случае отказа от запуска служб ViPNet необходимо будет запустить их вручную или перезагрузить компьютер.

Примечание. Согласно стандарту RFC-3704 (http://www.ietf.org/rfc/rfc3704.txt) в современных ОС Linux по умолчанию механизм фильтрации пакетов Unicast Reverse Path Forwarding Filtering включен в режиме Strict. Данный режим может вызывать конфликтные ситуации при работе ПО ViPNet. Поэтому при старте служб ViPNet фильтрация пакетов автоматически переводится в режим Loose.

В процессе работы программы установки ведется протокол (лог) установки, который находится в файле `/var/log/vipnet_install.log`, а также создается файл `/var/log/vipnet_files.txt`, в который записываются пути ко всем установленным файлам. В случае успешного завершения установки эти файлы копируются в каталог с установленной конфигурацией ViPNet.

Ручное конфигурирование ViPNet Coordinator Linux

Как описано выше, вы можете прервать процесс установки. Кроме того, после установки может возникнуть необходимость изменить конфигурацию. В этих случаях необходимо произвести ручное конфигурирование ViPNet Coordinator Linux с правами администратора (root). Ручное конфигурирование заключается в распаковке дистрибутива ключей (если он еще не распакован) и настройке параметров окружения.

Дистрибутив ключей представляет собой один файл `*.dst`, в котором содержатся адресные справочники и ключи.

Чтобы произвести ручное конфигурирование:

1. Выберите (и создайте, если нужно) каталог, где будут храниться справочники и ключи, и распакуйте дистрибутив в этот каталог командой `/sbin/unmerge <file.dst> <каталог>`.
2. Отредактируйте файл `/etc/iplirpsw`, который указывает местоположение справочников и ключей. Это текстовый файл, имеющий следующую структуру: первая строка – полный путь к каталогу, где находятся справочники и ключи, вторая строка – пароль доступа к этой информации, остальные строки игнорируются. После инсталляции этот файл содержит две строки с примером синтаксиса данного файла, эти строки нужно удалить и вписать свои.

> **Примечание.** Начиная с версии 3.5.2, пароль хранится в файле `/etc/iplirpsw` в зашифрованном виде в целях безопасности. Указывать пароль в открытом виде можно, но после успешной аутентификации пользователя пароль будет зашифрован и в таком виде перезаписан. При возврате к более ранней версии необходимо после установки более ранней версии вручную заменить зашифрованный пароль в файле `/etc/iplirpsw` на незашифрованный.

3. Отредактируйте файл `/etc/iplirnetpsw`, в котором задаются параметры доступа к защищенному сетевому узлу (см. «Программа просмотра информации о защищенном узле» на стр. 140).
4. После редактирования этих файлов (`/etc/iplirpsw` и `/etc/iplirnetpsw`) произведите установочный запуск управляющего демона командой `iplir check`. При этом управляющий демон не остается в памяти, а создает свои файлы конфигурации и завершает работу. Созданные файлы находятся внутри каталога, содержащего справочники и ключи, в подкаталоге `user`. Создается один основной файл, который называется `iplir.conf`, и по одному файлу конфигурации для каждого сетевого интерфейса, который присутствует на компьютере, файлы имеют имена `iplir.conf-eth0`, `iplir.conf-eth1` и так далее (см. «Общие принципы настройки» на стр. 48).

Примечание. Все манипуляции с управляющим демоном производятся с помощью скрипта `iplir`, который находится в каталоге `/sbin`. Рекомендуется добавить этот каталог в системную переменную PATH (если его там нет), тогда управляющий скрипт можно вызывать командой `iplir`. Если каталога `/sbin` нет в PATH, то скрипт `iplir` нужно вызывать командой `/sbin/iplir`.

Некоторые командные оболочки, например `tcsh`, кэшируют содержимое каталогов, содержащихся в PATH, поэтому сразу после установки команда `iplir` может не работать, хотя каталог `/sbin` содержится в PATH. В этом случае нужно набрать команду этой оболочки, которая обновляет кэш (например, `rehash` для `tcsh`). За более подробными инструкциями обратитесь к руководству по используемой командной оболочке.

5 Перед началом работы просмотрите созданные конфигурационные файлы и отредактируйте их так, чтобы информация в них соответствовала действительности (см. «Настройка параметров защищенного подключения» на стр. 47). При формировании дистрибутива ключей администратор задает IP-адреса и другие установки для каждого узла защищенной сети, но по недосмотру администратора информация может оказаться неверной или устаревшей. Если данный узел не единственный координатор в сети, то необходимо, чтобы были верно заданы адреса всех других координаторов, с которыми данному координатору разрешено соединяться. Подробнее редактирование файлов конфигурации описывается ниже.

6 Затем запустите управляющий демон командой `iplir start`. Эта команда также загружает драйвер ViPNet, если он не был загружен ранее. Демон запускается в фоновом режиме. Проверить его наличие в памяти можно командой `ps afx |grep -v grep |grep iplircfg |wc -l` (см. «Процессы-демоны, входящие в состав ПО ViPNet Linux» на стр. 169). Если эта команда выдает число больше нуля, то управляющий демон запущен.

Управляющий демон может не запуститься по нескольким причинам: испорченный дистрибутив, неправильно указанный пароль в файле `/etc/iplirpsw` и тому подобное. При запуске и последующей работе управляющий демон посылает информационные сообщения и сообщения об ошибках в `syslog`, используя `facility "daemon"` и `level "err"` для ошибок или `"info"` для информационных сообщений. Кроме того, при старте сообщения об ошибках дублируются на консоль. Обычно системный журнал находится в файле `/var/log/messages` или `/var/log/syslog`.

7 Если управляющий демон запустился, то проверьте функционирование защищенной сети, например, с помощью команды `ping`, посылая пакеты до какого-либо защищенного узла, о котором точно известно, что его адрес задан верно в файле `iplir.conf`. Если приходят ответные пакеты, то сеть функционирует нормально. Если не удается связаться с каким-либо защищенным узлом, то нужно отредактировать файлы конфигурации, как описано ниже.

8 Если все работает, перезагрузите систему и проверьте, запускается ли ПО ViPNet при загрузке. В случае успешной установки при загрузке запускаются драйверы ViPNet и демон системы защиты от сбоев failoverd (см. «Система защиты от сбоев» на стр. 126), который в свою очередь стартует управляющий демон (iplircfg) и другие демоны ViPNet. При этом система полностью готова к работе в защищенной сети без каких-либо действий со стороны администратора.

Дополнительная информация об изменениях, производимых в системе при установке ПО ViPNet Coordinator Linux, приведена в приложении Изменения, производимые в системе при установке и удалении ViPNet Coordinator Linux (на стр. 173).

Пересборка ядра Linux

Если при уже установленном ПО ViPNet Coordinator Linux решено пересобрать ядро Linux, то после установки нового ядра ПО ViPNet не будет работать и его необходимо будет переустановить. Примерный алгоритм пересборки ядра в таком случае следующий:

1. Сохраните файлы конфигурации ПО ViPNet Coordinator Linux (`/etc/iplirpsw`, `/etc/iplirnetpsw`, `/etc/failover.ini`, `/etc/iplirSKnums`).
2. Запустите скрипт `/sbin/rmiplir`, который удалит ПО ViPNet Coordinator Linux из системы.
3. Перезагрузите систему.
4. Пересоберите и установите ядро.
5. Перезагрузите компьютер с новым ядром.
6. Установите ПО ViPNet Coordinator Linux из дистрибутива.
7. Запишите в каталог `/etc` сохраненные файлы.
8. Перезагрузите компьютер с установленным ПО ViPNet Coordinator Linux.

Кроме того, при пересборке ядра необходимо иметь в виду, что при изменении некоторых важных опций (таких, как **Symmetric Multi-Processing Support**) недостаточно просто изменить соответствующую установку в меню конфигурации ядра и выполнить указанную выше последовательность команд для сборки ядра. В большинстве случаев при этом получается неработоспособное ядро. Даже выполнения команды `make clean` недостаточно для нормальной сборки. Примерный алгоритм сборки в этом случае следующий:

1. Сохраните файл `.config`, содержащий конфигурацию ядра
2. Выполните команду `make mrproper`.
3. Скопируйте сохраненный файл `.config` в каталог `/usr/src/linux/`.
4. Выполните команду `make menuconfig` и сделайте необходимые изменения.
5. Выполните указанную выше последовательность команд для сборки ядра.

Более подробно об этом см. в документации по ядру Linux.

Обновление версии ПО ViPNet Coordinator Linux

Внимание! Обновление ПО ViPNet Coordinator Linux до версии 4.1 возможно только с версий 3.6.x и 3.7.x.

Поэтому если у вас установлена более ранняя версия, чем 3.6.x, то сначала вам необходимо обновить ПО до версии 3.6.x или 3.7.x.

Во избежание потери удаленного доступа к компьютеру не рекомендуется выполнять обновление версии ПО в удаленной SSH-сессии.

Если необходимо заменить уже установленную на компьютере версию ПО ViPNet Coordinator Linux на более новую, выполните следующие действия:

1. Ознакомьтесь с особенностями конвертации правил открытой (см. «Конвертация правил открытой сети при обновлении до версии 4.x» на стр. 34) и защищенной сети (см. «Конвертация правил защищенной сети при обновлении до версии 4.x» на стр. 37) и при необходимости измените настройки правил перед обновлением ПО ViPNet Coordinator Linux.
2. Скопируйте на компьютер установочный файл новой версии.
3. Распакуйте установочный файл в любой каталог командой `tar xvfz <file>`. При распаковке будет создан каталог `distribute`.
4. Войдите в каталог `distribute`, где находится распакованный установочный файл, и выполните команду `./install.sh` (см. «Установка ПО ViPNet Coordinator Linux» на стр. 27).
5. Если в процессе работы программы установки будет предложено остановить запущенные службы ViPNet автоматически, то согласитесь с этим.
6. В процессе установки будет предложено сохранить текущую конфигурацию. Согласитесь с этим предложением.
7. После завершения установки будет предложено запустить службы ViPNet автоматически. Если необходимо, чтобы службы стартовали сразу после установки, то согласитесь с этим предложением.

Процесс обновления версии ПО ViPNet Coordinator Linux при работе узла в составе кластера горячего резервирования имеет ряд особенностей и подробно описан в документе «ViPNet Coordinator Linux. Система защиты от сбоев. Руководство администратора».

Примечание. При обновлении ПО ViPNet Coordinator Linux с версии 3.7 до версии 4.1 и выше журнал переключений кластера горячего резервирования будет перемещен в архив. После обновления в новом файле журнала переключений кластера содержится только информация с момента обновления. Если вы хотите просмотреть прежний журнал переключений, обратитесь в службу поддержки ОАО «ИнфоТеКС».

В результате ПО ViPNet Coordinator Linux будет обновлено.

Примечание. В связи с тем, что в ПО ViPNet Coordinator Linux версии 4.0 режимы безопасности не используются, то после обновления до версии 4.0 режимы безопасности преобразуются в соответствующие фильтры открытой и защищенной сетей (см. «Настройка параметров фильтрации IP-трафика» на стр. 77).

Конвертация правил открытой сети при обновлении до версии 4.x

В ПО ViPNet Coordinator Linux версий ниже 4.0 правила обработки открытых (нешифрованных) IP-пакетов содержатся в файле `firewall.conf`, и изменение правил производится редактированием данного конфигурационного файла. В версии 4.x правила фильтрации и правила трансляции IP-адресов хранятся в нескольких файлах `*.xml`, а их редактирование производится с помощью командного интерпретатора. Поэтому при обновлении ПО ViPNet Coordinator Linux до версии 4.x с более ранних версий происходит конвертация правил в новый формат.

Примечание. После успешной конвертации файл `firewall.conf` переименовывается в `firewall.conf.X.X.X(номер версии)`.

Если по каким-либо причинам при обновлении некоторые правила не будут конвертированы в новый формат, вы всегда сможете просмотреть правила, которые были заданы до обновления, и при необходимости добавить их самостоятельно (см. «Создание сетевого фильтра» на стр. 99).

При обновлении необходимо учитывать следующие особенности переноса правил:

- Во время конвертации правила открытой сети преобразуются в сетевые фильтры или правила трансляции и разбиваются на несколько таблиц (локальные фильтры, транзитные фильтры, фильтры для туннелируемых узлов, правила трансляции) в зависимости от того, к какой секции файла `firewall.conf` они относились (`[local]`, `[forward]`, `[tunnel]`, `[nat]`).

  Правила открытой сети из секции `[broadcast]` добавляются в таблицу локальных фильтров открытой сети, и для них в качестве адреса назначения указывается системная группа объектов BroadCast (см. «Системные группы объектов» на стр. 86).

- После конвертации правил в новый формат последовательность применения правил, заданная в текущей версии, сохраняется.
- Комментарии, заданные в правилах в текущей версии, не конвертируются в формат версии 4.x.
- Правила с масками идентификаторов узлов, отличающимися от маски сети (16), не конвертируются в формат версии 4.x. Перед обновлением необходимо привести такие правила к поддерживаемому формату.

- Если в правиле было задано расписание, то при конвертации расписание будет проигнорировано.
- Если в правиле заданы порты (в адресах источника и назначения) и не указан протокол TCP или UDP, такие правила не конвертируются в формат версии 4.x.
- Правила, отключенные в конфигурационном файле параметром `disable`, не конвертируются. Перед обновлением ПО ViPNet Coordinator Linux до версии 4.x необходимо предварительно включить выключенные правила.
- В версии ViPNet Coordinator Linux 3.x возможно формирование условия правил в виде `IP:Port`. Правило с подобным условием невозможно преобразовать в сетевой фильтр версии 4.x, поэтому при конвертации подобные правила разбиваются на несколько отдельных сетевых фильтров. То есть если в правиле в поле `from` содержится список из двух подобных условий, а в поле `to` – список из двух условий, то в результате конвертации данное правило будет преобразовано в четыре сетевых фильтра.

  Например, локальное правило фильтрации `rule= proto tcp from 192.168.1.1:80,192.168.1.5:2525 to 192.168.30.3:80, 192.168.4.2:443 pass` после конвертации будет выглядеть следующим образом:

  Таблица 4. Пример конвертации правил

Num	Name		Option
Act	Source	Destination	Protocol
1	RuleName1		User
pass	192.168.1.1	192.168.30.3	tcp: from 80 to 80
2	RuleName2		User
pass	192.168.1.1	192.168.4.2	tcp: from 80 to 443
3	RuleName3		User
pass	192.168.1.5	192.168.30.3	tcp: from 2525 to 80
4	RuleName4		User
pass	192.168.1.5	192.168.4.2	tcp: from 2525 to 443

- Если в условии правила для пропускания или блокировки пакетов указаны оба протокола (TCP, UDP) и списки портов источника и назначения, то после конвертации подобное условие будет разбито на несколько условий, которые будут попарно перечислять порты.

  Например, для правила `rule = proto tcp,udp from any:80,2525 to any:368,53 pass` после конвертации мы получим четыре фильтра, пропускающих или блокирующих пакеты:

Таблица 5. Пример конвертации правил, в которых указаны оба протокола (TCP, UDP)

Num	Name		Option
Act	Source	Destination	Protocol
1	RuleName1		User
pass	@any	@any	tcp: from 80 to 368,53
2	RuleName2		User
pass	@any	@any	tcp: from 2525 to 368,53
3	RuleName3		User
pass	@any	@any	udp: from 80 to 368,53
4	RuleName4		User
pass	@any	@any	udp: from 80 to 368,53

- Режимы безопасности конвертируются в локальные фильтры открытой сети. Так как для каждого сетевого интерфейса координатора задавался свой режим безопасности, то при конвертации создаются сетевые фильтры для каждого интерфейса.

  2-й и 5-й режимы безопасности не конвертируются. Поэтому для интерфейсов, которые до обновления были неактивны, сетевые фильтры будут отсутствовать, так как по умолчанию для неактивных интерфейсов задавался 2-й режим безопасности. Если после обновления требуется включить интерфейс, который был неактивен, то необходимые сетевые фильтры для этого интерфейса нужно добавить вручную.

  При конвертации режимов безопасности создаются следующие локальные фильтры открытой сети:

Таблица 6. Конвертация режимов безопасности

Num	Name		Option
Act	Source	Destination	Protocol
1	RuleName1		User
	(1 режим безопасности — блокировать IP-пакеты всех соединений)		
drop	@any [интерфейс]	@any [интерфейс]	@any
2	RuleName2		User
	(3 режим безопасности — пропускать все исходящие соединения кроме запрещенных)		

Num	Name		Option
Act	Source	Destination	Protocol
pass	@local [интерфейс]	@any	@any
3	RuleName3		User
	(4 режим безопасности — пропускать все соединения)		
pass	@any [интерфейс]	@any [интерфейс]	@any

Конвертация правил защищенной сети при обновлении до версии 4.x

В ПО ViPNet Coordinator Linux версий ниже 4.0 параметры правил защищенной сети содержатся в файлах `iplir.conf` и `iplir.serv`, изменение правил защищенной сети производится путем редактирования конфигурационного файла `iplir.conf`. В версии 4.x правила защищенной сети переименованы в фильтры защищенной сети, и редактирование данных фильтров производится с помощью командного интерпретатора. Поэтому при обновлении ПО ViPNet Coordinator Linux до версии 4.x с более ранних версий происходит конвертация правил защищенной сети в новый формат.

Примечание. После конвертации файлы `iplir.conf` и `iplir.serv` сохраняются в каталоге `user/config.X.X.X`.

Если по каким-либо причинам при обновлении некоторые правила не будут конвертированы в новый формат, вы всегда сможете просмотреть правила, которые были заданы до обновления, и при необходимости добавить их самостоятельно (см. «Создание сетевого фильтра» на стр. 99).

При обновлении необходимо учитывать следующие особенности переноса правил:

- Все правила защищенной сети во время конвертации преобразуются в соответствующие сетевые фильтры и добавляются в таблицу фильтров защищенной сети.
- Правила, отключенные в конфигурационном файле параметром `disable`, не конвертируются. Перед обновлением ПО ViPNet Coordinator Linux до версии 4.x необходимо предварительно включить выключенные правила.
- Правила, содержащие синтаксические ошибки, не конвертируются в новый формат.
- В фильтрах версии 4.x отсутствует атрибут направления пакета (например, `send`/`recv`/`any`), как в предыдущих версиях. Поэтому каждый фильтр с направлением пакета, равным `any`, конвертируется в два идентичных фильтра для входящих и исходящих пакетов.

- При конвертации все правила для широковещательного защищенного трафика преобразуются в сетевые фильтры в соответствии с указанным действием (`drop` или `pass`), и дополнительно создается общий широковещательный фильтр с противоположным действием.
- При конвертации все совпадающие правила для различных узлов объединяются. При этом узлы объединяются в группы, для которых создаются соответствующие сетевые фильтры.
- Правила для защищенных узлов преобразуются следующим образом:
  - Если в правиле для группы узлов был установлен разрешающий фильтр по умолчанию и в правило были добавлены блокирующие фильтры, то при конвертации создаются блокирующие фильтры, а разрешающий фильтр по умолчанию не создается.
  - Если в правиле для группы узлов установлен блокирующий фильтр по умолчанию и в правило были добавлены разрешающие фильтры, то при конвертации создаются разрешающие фильтры и блокирующий фильтр по умолчанию, который имеет наименьший приоритет в списке фильтров.
- Преобразование главного фильтра защищенной сети осуществляется следующим образом:
  - Если в правиле «IP-пакеты всех адресатов» (в главном фильтре) установлен блокирующий фильтр по умолчанию и добавлены разрешающие фильтры, то при конвертации в конец таблицы фильтров добавляется эти разрешающие фильтры.
  - Если в правиле «IP-пакеты всех адресатов» установлен блокирующий фильтр по умолчанию и отсутствуют дополнительные фильтры, то при конвертации в начало таблицы фильтров добавляется блокирующий все соединения фильтр.
  - Если в правиле «IP-пакеты всех адресатов» установлен разрешающий фильтр по умолчанию и добавлены блокирующие фильтры, то при конвертации в конец таблицы добавляется разрешающий все соединения фильтр, а блокирующие фильтры добавляются в начало таблицы.
  - Если в правиле «IP-пакеты всех адресатов» установлен разрешающий фильтр по умолчанию и отсутствуют дополнительные фильтры, то при конвертации в конец таблицы фильтров добавляется разрешающий все соединения фильтр.

Рассмотрим пример конвертации главного фильтра защищенной сети (разрешающий фильтр по умолчанию и дополнительные блокирующие фильтры). До конвертации в файле `iplir.conf` указаны следующие параметры:

```
[id]
id= 0xfffffffe
name= MainFilter
filterdefault= pass
filtertcp= <port> drop, send

[id]
id= <ID1>
name= Coordinator_1_1
filterdefault= drop
filtertcp= <port> pass, send

[id]
```

```
id= <ID2>
name= Station_1_1_1
filterdefault= pass
filtertcp= <port> drop, send
```

Таблица 7. Таблица фильтров защищенной сети после конвертации

Num	Name		Option
Act	Source	Destination	Protocol
1	MainFilter		User
drop	local	@any	tcp: from <port>
2	RuleName2		User
pass	local	<ID1>	tcp: from <port>
3	RuleName3		User
drop	local	<ID1>	
4	RuleName4		User
drop	<ID1>	@any	tcp: from <port>
5	RuleName5		User
drop	local	<ID2>	tcp: from <port>
6	MainFilter		User
pass	@any	@any	

Переход от использования фиксированных альтернативных каналов к множественным адресам доступа

Если раньше вы использовали ViPNet Coordinator Linux версии 4.0 и ниже и для каких-либо узлов было настроено использование фиксированных альтернативных каналов, после обновления программного обеспечения необходимо выполнить аналогичные настройки с помощью множественных адресов доступа и их метрик, так как настройки фиксированных альтернативных каналов не конвертируются при обновлении.

При настройке необходимо учитывать следующие особенности:

- Настройки фиксированных альтернативных каналов больше не выполняются (секция `[channels]` теперь не используется).
- Группы узлов больше не создаются (параметр `group` в секциях `[id]` теперь не используется).

- Внешний адрес узла, который использовался для связи через фиксированный альтернативный канал (параметр `channelfirewallip`), теперь указывается в секции `[id]` этого узла в качестве одного из адресов доступа к узлу (параметр `accessiplist`).
- Приоритет использования того или иного адреса доступа к узлу задается с помощью метрики в параметре `accessiplist`.
- Раньше для узлов могли назначаться различные порты доступа при использовании фиксированных альтернативных каналов связи (параметр `channelport` в секциях `[id]`), теперь для всех каналов связи с узлом используется один порт (параметр `port` в секциях `[id]`).

Удаление ViPNet Coordinator Linux

Если возникла необходимость удалить ViPNet Coordinator Linux с компьютера, то нужно воспользоваться скриптом `rmiplir`, который помещается после установки в каталог `/sbin`.

Чтобы удалить ПО ViPNet Coordinator Linux, выполните следующие действия:

1. Выгрузите службы и драйверы ViPNet вручную командой `vipnet stop`.
2. Запустите скрипт `rmiplir` с параметром YES (`rmiplir YES`), что исключает возможность случайного удаления ViPNet Coordinator Linux.

   Скрипт `rmiplir` удалит ПО ViPNet, файлы конфигурации в каталоге `/etc`, файлы устройств в каталоге `/dev` и другие файлы.
3. Для удаления справочников и ключей вручную выполните команду `rm`.
4. Удалите сам скрипт `rmiplir` из каталога `/sbin` вручную.
5. После удаления ПО ViPNet Coordinator Linux перезагрузите компьютер. Если ПО было выгружено вручную, то выполнять перезагрузку не обязательно.

3

Работа с ViPNet Coordinator Linux

Управление ViPNet Coordinator Linux 43

Командный интерпретатор ViPNet Coordinator Linux 44

Обновление справочников и ключей 46

Управление ViPNet Coordinator Linux

Практически все функции, необходимые для управления ViPNet Coordinator Linux, могут быть выполнены с помощью скрипта `iplir`, расположенного в каталоге `/sbin`. Возможны следующие команды:

- `iplir start` – запускает управляющий демон. Если при этом не существуют файлы конфигурации, то они создаются, как и по команде `iplir check`.
- `iplir stop` – останавливает управляющий демон, при этом драйвер ViPNet продолжает работать.
- `iplir unload` – останавливает управляющий демон и выгружает драйвер ViPNet.

  Перед выполнением данной команды появляется предупреждение о том, что в результате компьютер окажется незащищенным и безопасность данных в этом случае не гарантируется. Затем запрашивается пароль пользователя данного сетевого узла для подтверждения выполнения команды. Команда выполняется только в случае ввода правильного пароля пользователя.
- `iplir check` – создает конфигурационные файлы, а также проверяет их синтаксис, если файлы уже существуют.
- `iplir info` – показывает информацию о локальном или удаленном защищенном узле (см. «Программа просмотра информации о защищенном узле» на стр. 140).
- `iplir view` – позволяет просматривать журнал трафика (см. «Программа просмотра журнала регистрации IP-пакетов» на стр. 134) локального или удаленного защищенного узла.
- `iplir passwd` – позволяет сменить пароль пользователя ViPNet (см. «Программа смены пароля пользователя» на стр. 144).
- `iplir config [команда]` – позволяет производить сохранение, восстановление, удаление и просмотр списка конфигураций ViPNet (см. «Программа работы с конфигурациями ViPNet» на стр. 146).

Запуск или остановка всех демонов и загрузка или выгрузка всех драйверов ViPNet производится с помощью скрипта `vipnet`, расположенного в каталоге `/sbin`. Возможны следующие команды:

- `vipnet stop` – останавливает все демоны и выгружает все драйверы ViPNet. Затем запрашивается пароль пользователя данного сетевого узла для подтверждения выполнения команды. Команда выполняется только в случае ввода правильного пароля пользователя.
- `vipnet start` – загружает все драйверы ViPNet и запускает все демоны.

Командный интерпретатор ViPNet Coordinator Linux

Командный интерпретатор предназначен для работы:

- с сетевыми фильтрами, которые используются в ViPNet Coordinator Linux для разрешения и блокирования входящего и исходящего трафика;
- с правилами трансляции IP-адресов;
- с опциями ПО ViPNet Coordinator Linux;
- с настройками обработки прикладных протоколов.

Также командный интерпретатор предоставляет функции разграничения доступа к настройкам программы ViPNet Linux Coordinator.

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

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 символ «>» в приглашении интерпретатора). В режиме пользователя недоступны некоторые команды, требующие прав администратора.

Для управления командным интерпретатором используются следующие команды:

- `vipnet shell` – вызов интерпретатора в командной консоли Linux. При выполнении данной команды потребуется ввести ваш пароль пользователя ViPNet.
- `enable` – вход в режим администратора. При выполнении данной команды необходимо ввести пароль администратора сетевого узла, заданный в программе ViPNet Центр управления сетью.
- `exit` – выход из текущего режима командного интерпретатора. При выполнении данной команды в режиме администратора осуществляется выход в режим пользователя, при выполнении в режиме пользователя — завершение работы командного интерпретатора.

В командном интерпретаторе выполняются команды следующих групп:

- `firewall` — команды данной группы предназначены для создания и редактирования сетевых фильтров, правил трансляции адресов и групп объектов (см. «Команды группы firewall» на стр. 194).
- `iplir option` — команды данной группы предназначены для настройки функции антиспуфинга и дополнительных опций межсетевого экрана (см. «Команды группы iplir option» на стр. 195).
- `alg` — команды данной группы предназначены для настройки параметров обработки прикладных протоколов (см. «Команды группы alg» на стр. 196).

Внимание! Команды по управлению демонами (`start`, `stop`, `info`, `view` и так далее) выполняются в командной консоли Linux.

Обновление справочников и ключей

Если администратор сети ViPNet вносит какие-либо изменения в структуру сети или настройки отдельных сетевых узлов, например, создает новые связи между сетевыми узлами, то автоматически изменяются справочники и ключи для сетевых узлов. Обновления справочников и ключей создаются администратором сети с помощью ПО ViPNet Administrator и могут быть автоматически отправлены на сетевые узлы, которых коснулись изменения.

На координаторе с установленным ПО ViPNet Coordinator Linux поступившие обновления справочников и ключей принимаются в автоматическом режиме.

Если по каким-либо причинам обновление справочников и ключей не может быть принято по сети, вы можете выполнить обновление вручную с помощью дистрибутива ключей. Для этого:

1. Получите новый дистрибутив ключей у администратора сети ViPNet и скопируйте его на компьютер.
2. Остановите все службы ViPNet.
3. Распакуйте новый дистрибутив ключей с помощью команды

   ```
   /sbin/unmerge -f <файл дистрибутива> <каталог>,
   ```

   где `<каталог>` — каталог, в котором хранятся текущие справочники и ключи. При этом текущие справочники и ключи будут автоматически удалены, прочие данные останутся в указанном каталоге (см. «Программа распаковки дистрибутива ключей» на стр. 145).
4. Запустите службы ViPNet.

4

Настройка параметров защищенного подключения

Общие принципы настройки 48

Настройка параметров защищенной сети 49

Настройка параметров сетевых интерфейсов 66

Настройка режимов работы через межсетевой экран 68

Настройка координатора, выполняющего функции сервера открытого Интернета 74

Общие принципы настройки

Настройка ViPNet Coordinator Linux производится путем редактирования файлов конфигурации `iplir.conf`, `iplir.conf-<интерфейс>`, сетевых фильтров открытой и защищенной сети, а также правил трансляции IP-адресов в командной консоли Linux. 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 другие специальные символы. Строка с именем секции считается началом секции. Секция заканчивается там, где начинается следующая секция, или в конце файла. Все имена секций, параметров, названия протоколов, 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 запуске управляющего демона или в процессе его работы и служат для информирования пользователя о некритических ошибках конфигурации, о значении фильтров по каким-либо сервисам и так далее. Вам не следует добавлять служебные комментарии самостоятельно, они будут потеряны после следующего запуска 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.

Настройка параметров защищенной сети

Параметры настройки защищенной сети содержатся в файле `iplir.conf`. Секции, которые может содержать этот файл, описаны 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 (собственная секция).

Секция `[id]` содержит следующие параметры:

- `id` — уникальный идентификатор сетевого узла. По этому параметру управляющий демон отличает одну секцию `[id]` от другой. Идентификатор присваивается сетевому узлу ViPNet при его создании, 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`.
- `ip` — IP-адрес данного узла. 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 Принципы назначения виртуальных адресов (на стр. 64).

  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, то есть виртуальный адрес перемещается вслед за своим реальным. Если указан только реальный адрес (запятой в этом случае нет), то считается, что этому адресу еще не сопоставлен виртуальный.

Внимание! Менять виртуальный адрес вручную запрещено, он назначается автоматически.

- `accessip` — текущий IP-адрес доступа к данному узлу со стороны собственного узла, то есть тот IP-адрес, который в текущий момент используется 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 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 запрещено, 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. 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 так далее) в этой же секции (см. ниже). Использование данного параметра для собственной секции `[id]` описано в разделе Настройка режимов работы через межсетевой экран (на стр. 68). 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.
- `accessiplist` — для всех секций `[id]` данный параметр определяет IP-адреса доступа к сетевому узлу и их приоритет (если узел имеет множественные адреса доступа). В каждой секции `[id]` может быть указано любое количество параметров `accessiplist` (по количеству адресов доступа узла). В первом параметре `accessiplist` в каждой секции в качестве адреса доступа должен быть указан тот же адрес, что и в параметре `firewallip`. Если параметры `accessiplist` будут отсутствовать в секции, то параметр `firewallip` тоже будет отсутствовать. Остальные параметры `accessiplist` в секции используются для формирования списка адресов доступа к узлу с данного координатора ViPNet Coordinator Linux.

  Значение параметра `accessiplist` указывается в виде:

  ```
  accessiplist= X.X.X.X, M, Y.Y.Y.Y, N, somestr
  ```

  где:

  - `X.X.X.X` — IP-адрес доступа к узлу. В частном случае может принимать нулевое значение, это означает, что данный узел (клиент) не находится за межсетевым экраном.
  - `M` — метрика данного адреса доступа, диапазон допустимых значений — 0-9999. Метрика определяет задержку (в миллисекундах) отправки тестовых пакетов при выполнении процедуры определения адреса доступа узла. Опросы осуществляются периодически (секция `[misc]`, параметры `server_pollinterval` и `client_pollinterval`). При назначении метрики нужно придерживаться следующих принципов:
    - Соединение устанавливается по тому адресу, доступность которого быстрее определяется в результате опроса.

- Адрес с наименьшей метрикой считается самым приоритетным. Соединение с узлом устанавливается по адресу с наименьшей метрикой всегда, когда этот адрес доступен.
- Значение метрики должно быть больше, чем максимальное время задержки при прохождении служебного сообщения и ответа на него по соответствующему каналу.
- По умолчанию значение метрики равно `auto` (метрика не присвоена). В этом случае метрика определяется автоматически. При наличии других записей с метриками в данной секции `[id]` значение метрики `auto` всегда будет на 100 миллисекунд больше максимального значения заданной метрики (то есть канал с автоматической метрикой всегда будет иметь самый низкий приоритет по сравнению с каналами, которым метрика назначена вручную).
- Рекомендуемая минимальная разница между метриками — 100. Чем больше разница между наименьшей метрикой и остальными метриками, тем меньше вероятность того, что в случае кратковременного сбоя самого приоритетного канала во время процедуры определения адреса доступа будет выбран менее приоритетный канал.
- Если все метрики равны, то для работы будет выбран тот канал, через который соединение с узлом будет установлено быстрее. После того как канал выбран, определение доступности других каналов связи выполняется только при потере соединения по текущему каналу. Этот же механизм действует в случае, если выбран канал связи с наименьшей метрикой.

o `Y.Y.Y.Y` — реальный адрес узла, соответствующий тому сетевому интерфейсу, через который будут передаваться IP-пакеты для данного адреса доступа.

o `N` — условный номер сетевого интерфейса, возможные значения — 1-255 или 0 (не определен). Данный параметр вычисляется системой, при добавлении адреса доступа вручную необходимо указывать номер интерфейса равный 0.

o `somestr` — тип регистрации данного адреса доступа узла. Может принимать следующие значения:

- `auto` — адрес доступа был задан ПО ViPNet Coordinator Linux.
- `manual` — адрес был задан администратором вручную (при редактировании `iplir.conf`).
- `addrdoc` — адрес доступа получен из справочников, присланных из программы ViPNet Центр управления сетью.
- `other` — адрес доступа, добавленный другим способом. Например, в качестве адреса доступа был добавлен координатор, выбранный внешним межсетевым экраном.

Параметр `somestr` вычисляется системой автоматически, при добавлении адреса доступа вручную необходимо указывать значение `manual`.

Примечание. Возможен сокращенный ввод значения параметра `accessiplist`. Вы можете указать только адрес доступа или адрес доступа и метрику, остальные значения будут подставлены системой автоматически при запуске управляющего демона.

- `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. 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. 68).

- `proxyid` — тип используемого режима работы данного узла через межсетевой экран. 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 в разделе Настройка режимов работы через межсетевой экран (на стр. 68). 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, чтобы обеспечить пропуск входящего трафика через межсетевой экран (см. «Настройка режима „С динамической трансляцией адресов“» на стр. 71). 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, то есть если параметр установлен 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. 68).

  Внимание! Для всех секций, кроме собственной, параметр `usefirewall` определяется автоматически. В этом случае менять параметр вручную не следует.

- `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. 68).

- `virtualip` — базовый виртуальный адрес данного узла. Назначение данного параметра описано в разделе Принципы назначения виртуальных адресов (на стр. 64). Каждая секция `[id]` имеет только один параметр `virtualip`.

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

- `visibility` – позволяет настроить тип видимости узла. Этот параметр может принимать следующие значения:
  - `auto` — автоматическое определение типа видимости узла, в зависимости от текущего адреса видимости узла.
  - `real` — позволяет всегда обращаться к данному узлу по его реальному адресу.
  - `virtual` — позволяет всегда обращаться к данному узлу по его виртуальному адресу.

  Этот параметр не является обязательным и используется, только если для данного узла необходимо задать индивидуальную настройку видимости. В случае отсутствия параметра `visibility` видимость узла определяется по значениям параметров, заданных в секции `[visibility]` (см. «Секция [visibility]» на стр. 62). В данной секции видимость узла определяется по параметрам доступа сети, которой принадлежит узел, либо по параметрам доступа узлов по умолчанию. Определение видимости узлов более подробно описано в разделе Правила определения видимости сетевых узлов (на стр. 63).

  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 `visibility` в значение `real`, 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. 71). 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.

- `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. «Программа просмотра журнала регистрации IP-пакетов» на

стр. 134). 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.

- `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 формируется 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.

  Обычно параметры `tunnel` рассылаются в составе справочников и ключей. Если туннелируемые адреса координатора заданы в ЦУСе, то другие узлы получат информацию об этом автоматически. Если туннелируемые адреса заданы на координаторе вручную, эти адреса также необходимо указать вручную на каждом узле, который будет работать с этими туннелируемыми узлами посредством сети ViPNet.

  Подробные примеры настройки туннелей в типовых схемах приведены в Приложении (см. «Примеры настройки туннелей с использованием координаторов ViPNet» на стр. 188).

- `exclude_from_tunnels` — исключение из туннелируемых адресов, заданных параметрами `tunnel`. Параметр используется в секциях `[id]`, которые описывают настройки координаторов, связанных с ViPNet Coordinator Linux. Задается в виде `ip1-ip2`, где `ip1` и `ip2` — начальный и конечный адреса диапазона, который не надо туннелировать. В секции `[id]` можно задать несколько параметров `exclude_from_tunnels`.

  Например, чтобы исключить адрес 192.168.201.7 из туннелируемого диапазона 192.168.201.5-192.168.201.10, то есть не шифровать трафик при соединении с этим адресом, необходимо задать строку:

  ```
  exclude_from_tunnels= 192.168.201.7-192.168.201.7
  ```

- `usetunnel` — включение или отключение туннелирования незащищенных компьютеров. Параметр используется в секциях `[id]`, которые описывают настройки координаторов, связанных с ViPNet Coordinator Linux. Может принимать значение `on` или `off`, значение по умолчанию — `on`. Если параметр имеет значение `off`, то при соединении ViPNet Coordinator Linux с узлами, которые туннелирует данный координатор, трафик не шифруется.

Примечание. Параметры `exclude_from_tunnels` и `usetunnel` задаются только локально и не изменяются при приеме обновлений справочников из программы ViPNet Центр управления сетью.

Некоторые из секций `[id]` имеют специальное значение. Самая первая секция `[id]` описывает компьютер, на котором установлен ViPNet Coordinator Linux. 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. Описание настроек различных режимов функционирования собственного узла через межсетевой экран приведено в разделе Настройка режимов работы через межсетевой экран (на стр. 68).
- 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 автоматически приводится в соответствие с реальным диапазоном.

С помощью параметров собственной секции `[id]` также настраиваются различные режимы работы ViPNet Coordinator Linux. 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 шлюза по умолчанию.
- Если собственный узел будет работать хотя бы с одним узлом по виртуальному адресу, то у него должен быть настроен шлюз по умолчанию. Если шлюз по умолчанию не будет указан, то пакет может быть блокирован 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-форвардинга, то есть параметр `ipforwarding` в секции `[misc]` должен быть установлен в значение `on`.

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 драйвер и управляющий демон периодически получают информацию о состоянии параметров известных им адаптеров с интервалом времени, задаваемым параметром `ifcheck_timeout` секции `[misc]`. Если обнаруживается, что адаптер деактивирован в системе, то он деактивируется и в драйвере сетевой защиты ViPNet. Если адаптер активируется или изменяется его адрес, то эти изменения автоматически загружаются в драйвер. Информация обо всех описанных событиях выводится в системный журнал.

В секции `[adapter]` указываются следующие параметры:

- `name` – системное имя адаптера, например: eth0, ppp0.

  Если в системе заданы несколько IP-адресов на одном адаптере и присутствуют одно или несколько псевдоустройств (eth0:0, eth0:1 и так далее), 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 узел не стоит за внешним межсетевым экраном или за ViPNet-координатором, использующимся как прокси-сервер, то есть работает в режиме **Без использования межсетевого экрана** (см. «Настройка режима „Без использования межсетевого экрана"» на стр. 68), то типы всех адаптеров должны быть установлены в `internal`.
  - Если узел стоит за внешним межсетевым экраном или за ViPNet-координатором, использующимся как прокси-сервер, то есть работает в режиме **Координатор** или **Со статической трансляцией адресов** (с фиксированным внешним адресом), то тип адаптера, посредством которого данный узел будет связываться с узлом, выполняющим функции межсетевого экрана, устанавливается в `external`, типы остальных адаптеров — в `internal`.

- `allowtraffic` — разрешение или блокирование прохождения трафика через данный интерфейс. Может принимать значение `on` или `off`. По умолчанию значение параметра `on`.

  Если параметр `allowtraffic` установлен в значение `on`, то пакеты, проходящие через интерфейс, фильтруются в соответствии с заданными сетевыми фильтрами и могут быть либо пропущены, либо блокированы. Если этот параметр установлен в значение `off`, то все пакеты будут блокироваться независимо от остальных настроек.

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 в разделе Настройка параметров сетевых интерфейсов (на стр. 66).

Секция [dynamic]

Секция `[dynamic]` содержит параметры, необходимые для настройки режима **С динамической трансляцией адресов** (см. «Настройка режима „С динамической трансляцией адресов"» на стр. 71):

- `dynamic_proxy` – включение или выключение режима **С динамической трансляцией адресов** на собственном узле. Этот параметр может принимать значение `on` или `off`, что соответствует включению или выключению данного режима. По умолчанию значение параметра `off`.

  В предыдущих версиях ViPNet Coordinator Linux данный параметр находился в секции `[misc]`, поэтому при переходе на текущую версию он автоматически переносится в секцию `[dynamic]`, о чем записывается соответствующее предупреждение в системном журнале. Описание выбора различных режимов для собственного узла приведено в разделе Настройка режимов работы через межсетевой экран (на стр. 68).

- `firewallip` – внешний IP-адрес доступа к собственному узлу, работающему в режиме **С динамической трансляцией адресов**, со стороны других сетевых узлов.

  В предыдущих версиях ViPNet Coordinator Linux данный параметр находился в секции `[id]` для собственного узла и назывался `dynamic_firewallip`, поэтому при переходе на текущую версию он автоматически переносится в секцию `[dynamic]`, о чем записывается соответствующее предупреждение в системном журнале.

  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**.

  В предыдущих версиях ViPNet Coordinator Linux данный параметр находился в секции `[id]` для собственного узла и назывался `dynamic_port`, поэтому при переходе на текущую версию он автоматически переносится в секцию `[dynamic]`, о чем записывается соответствующее предупреждение в системном журнале.

  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. 71).

- `always_use_server` — включение или выключение режима, при котором любой трафик с внешними узлами направляется через координатор, выбранный в параметре `forward_id` данной секции, то есть 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. 71).
- `timeout` — период опроса (в секундах) координатора для обеспечения пропуска входящего трафика через межсетевой экран (по умолчанию — 25 секунд). Описание установки данного параметра приведено в разделе Настройка режима «С динамической трансляцией адресов» (на стр. 71).

Секция [misc]

Секция `[misc]` описывает 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`.

  Примечание. Установка формата `4.0` совместно с алгоритмом шифрования AES не допускается (см. описание параметра `ciphertype`).

  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 устанавливать соединения 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 может принимать следующие значения:
  - `gost` — шифрование с помощью алгоритма ГОСТ;
  - `aes` — шифрование с помощью алгоритма AES.

  По умолчанию используется значение `gost`.

  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`. Значение параметра должно быть больше 1 секунды и меньше 7200 секунд включительно. По умолчанию его значение равно 7200 (то есть два часа).
- `tunnel_local_network` — параметр, который позволяет не туннелировать IP-адреса, входящие в подсеть ViPNet Coordinator Linux. Может принимать следующие значения:
  - `on` — обращаться к туннелируемым узлам, находящимся в локальной подсети, через координатор, который туннелирует данные узлы. При данном значении параметра доступ к туннелируемым узлам в локальной подсети может быть затруднен.
  - `off` — обращаться к туннелируемым узлам, находящимся в локальной подсети, минуя туннелирующий координатор. Данное значение используется по умолчанию.
- `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 прийти 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.
- `iscaggregate` — накопление служебного трафика, обрабатываемого на координаторе (данная функция недоступна на клиенте). Может принимать следующие значения:
  - `on` — накопление служебного трафика в течение минуты с последующей рассылкой на узлы не чаще, чем 1 раз в минуту (установлено по умолчанию);
  - `off` — накопление служебного трафика отключено.

  Включение опции `iscaggregate` позволяет уменьшить объем служебного трафика.
- `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.

- `ifcheck_timeout` — интервал опроса (в секундах) параметров адаптеров, известных управляющему демону. По умолчанию значение данного параметра равно 30 секунд.
- `warnoldautosave` — включение или выключение предупреждения о наличии старых автоматически сохраненных конфигураций ViPNet (см. «Программа работы с конфигурациями ViPNet» на стр. 146). 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.
- `mssdecrease` — число байтов, на которое будет уменьшен параметр MSS (максимальный размер сегмента) протокола TCP (см. «Фрагментирование шифрованных ViPNet-пакетов» на стр. 153) для исключения фрагментации шифрованных ViPNet-пакетов. Значение по умолчанию — `0`. В случае если проверка соединения между узлами (ping) или туннелируемыми узлами 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.

- `msg_compress_level` — уровень сжатия служебных межсерверных сообщений. Может принимать значение от 1 (минимальная компрессия, максимальная скорость) до 9 (максимальная компрессия, минимальный объем служебного трафика). Значение по умолчанию — 3.

Примечание. На высоконагруженных узлах не рекомендуется устанавливать значение параметра `msg_compress_level` больше 5.

Секция [servers]

Секция `[servers]` содержит список координаторов, известных данному сетевому узлу. 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 [virtualip]

Секция `[virtualip]` содержит установки виртуальных адресов (см. «Принципы назначения виртуальных адресов» на стр. 64). 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` узлов.
- `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 3.6.4. При обновлении более ранней версии на версию 3.6.4 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. 159).

Секция [visibility]

Секция `[visibility]` содержит настройки видимости защищенных сетевых узлов, с которыми связан ViPNet Coordinator Linux. В отличие от параметра `visibility`, с помощью которого в секциях `[id]` задается видимость отдельных узлов, в этой секции можно задать видимость сразу для всех узлов сетей или подсетей ViPNet. Настройки, заданные в секции `[visibility]`, учитываются при определении видимости узлов со стороны собственного узла (см. «Правила определения видимости сетевых узлов» на стр. 63).

Секция может содержать следующие параметры:

- `default` — видимость узлов по умолчанию. Этот параметр является обязательным и может принимать следующие значения:
  - `auto` — автоматически определять видимость узлов;
  - `real` — для доступа к узлам использовать их реальные IP-адреса;
  - `virtual` — для доступа к узлам использовать их виртуальные IP-адреса (см. «Виртуальный IP-адрес» на стр. 210).

  По умолчанию значение параметра — `auto`.
- `subnet_real` — идентификаторы сетей ViPNet, для которых необходимо настроить видимость узлов по реальным IP-адресам.

  Идентификаторы указываются в шестнадцатеричном формате. В секции `[visibility]` можно задать несколько параметров `subnet_real`. В одном параметре можно указать либо один идентификатор, либо несколько идентификаторов, разделенных запятой. Например:

  ```
  subnet_real= 0x5155
  subnet_real= 0x5156, 0x5157, 0x5158
  ```
- `subnet_virtual` — идентификаторы сетей ViPNet, для которых необходимо настроить видимость узлов по виртуальным IP-адресам.

  Задается так же, как параметр `subnet_real`. В секции `[visibility]` можно задать несколько параметров `subnet_virtual`.

  Параметры `subnet_real` и `subnet_virtual` являются необязательными и по умолчанию отсутствуют в секции `[visibility]`.

Внимание! Один и тот же идентификатор сети можно указать только в одном из параметров `subnet_real` или `subnet_virtual`.

При старте управляющего демона идентификаторы, заданные в параметрах `subnet_real` и `subnet_virtual`, автоматически сортируются в порядке возрастания и группируются максимум по 8 идентификаторов в одной строке.

Правила определения видимости сетевых узлов

Для связи ViPNet Coordinator Linux с узлами, описанными в секциях `[id]`, могут использоваться реальные или виртуальные адреса. При этом для связи с узлами, от которых ViPNet Coordinator Linux получает широковещательные пакеты (broadcast), используются их реальные адреса. В иных ситуациях используемый тип адреса для связи с узлом определяется следующим образом:

- Если ViPNet Coordinator Linux работает в режиме **Без использования межсетевого экрана** (см. «Настройка режима „Без использования межсетевого экрана“» на стр. 68), то есть не стоит за внешним межсетевым экраном, то:
  - Для связи с клиентами, работающими в режиме **Без использования межсетевого экрана**, то есть не стоящими за какими-либо внешними межсетевыми экранами, используются реальные адреса.
  - Для связи с узлами, работающими в режиме **Координатор** и использующими ViPNet Coordinator Linux как прокси-сервер, используются реальные адреса.
  - Для связи со всеми другими сетевыми узлами используются виртуальные адреса.
- Если ViPNet Coordinator Linux работает в режиме **Координатор**, то есть стоит за каким-либо ViPNet-координатором, выполняющим функции прокси-сервера, то:
  - Для связи с клиентами, работающими в режиме **Без использования межсетевого экрана**, то есть не стоящими за какими-либо внешними межсетевыми экранами, используются реальные адреса.
  - Для связи с координаторами, кроме ViPNet-координатора, используемого как прокси-сервер, всегда используются виртуальные адреса.
  - Для связи с сетевыми узлами, стоящими за тем же интерфейсом того же самого ViPNet-координатора, что и ViPNet Coordinator Linux (их `firewallip` равен `firewallip`, заданному в настройках ViPNet Coordinator Linux), а также для связи с самим ViPNet-координатором используются реальные адреса.
  - Для связи с сетевыми узлами, стоящими за тем же самым ViPNet-координатором, что и ViPNet Coordinator Linux, но за другим интерфейсом, а также работающими в режиме, отличном от режима **Без использования межсетевого экрана**, то есть стоящими за какими-либо внешними межсетевыми экранами, используются виртуальные адреса.
- Если ViPNet Coordinator Linux работает в режимах **Со статической трансляцией адресов** или **С динамической трансляцией адресов** (см. «Настройка режима „С динамической трансляцией адресов“» на стр. 71), то есть стоит за каким-либо внешним межсетевым экраном, то:
  - Для связи с клиентами, работающими в режиме **Без использования межсетевого экрана**, то есть 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 ViPNet Coordinator Linux (их `firewallip` равен `firewallip`, заданному в настройках ViPNet Coordinator Linux), используются реальные адреса.
- Для связи со всеми другими сетевыми узлами используются виртуальные адреса.

Приведенные выше правила определения видимости узлов применяются автоматически, однако их можно изменить с помощью параметра `visibility` в секциях `[id]` (см. «Секция [id]» на стр. 49) и с помощью настроек в секции `[visibility]` (см. «Секция [visibility]» на стр. 62). Видимость каждого узла с учетом заданных настроек определяется следующим образом:

1. Если в секции `[id]`, описывающей узел, присутствует параметр `visibility`, то видимость узла определяется по значению этого параметра: `real` — использовать реальный адрес, `virtual` — использовать виртуальный адрес, `auto` — определять видимость автоматически.
2. Если в секции `[id]` узла нет параметра `visibility`, но при этом сеть, которой принадлежит узел, указана в одном из параметров `subnet_real` или `subnet_virtual` секции `[visibility]`, то видимость узла определяется по названию параметра: `subnet_real` — использовать реальный адрес, `subnet_virtual` — использовать виртуальный адрес.
3. В иных случаях видимость узла определяется по значению параметра `default` секции `[visibility]`: `real` — использовать реальный адрес, `virtual` — использовать виртуальный адрес, `auto` — определять видимость автоматически.

Таким образом, индивидуальные настройки видимости узлов имеют приоритет над настройками видимости сетей, а настройки видимости сетей — над настройкой видимости по умолчанию.

Принципы назначения виртуальных адресов

Виртуальные адреса используются для сетевых узлов, которые подключаются к внешней сети через межсетевой экран. 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.

Параметры, необходимые для назначения виртуальных адресов, задаются администратором в файле `iplir.conf` в секции `[virtualip]` (см. «Секция [virtualip]» на стр. 61).

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

- Старший октет всех виртуальных адресов имеет одинаковое значение, соответствующее старшему октету стартового виртуального адреса `startvirtualip`.

Примечание. Если при назначении виртуальных адресов все адреса с одинаковым старшим октетом были исчерпаны, старший октет будет увеличен на единицу. Например: 11.255.255.254, 12.0.0.1, 12.0.0.2...

- Два младших октета характеризуют сетевой узел. Они одинаковы для всех виртуальных адресов данного сетевого узла и различны для виртуальных адресов, принадлежащих разным сетевым узлам. Значения двух младших октетов не превышают соответствующих значений, заданных максимальным виртуальным адресом `maxvirtualip`.
- Второй по старшинству (слева) октет характеризует один из реальных адресов сетевого узла.

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

При появлении каждого сетевого узла в списке связей ему назначается виртуальный адрес, который привязан к уникальному идентификатору этого узла и соответствует его первому реальному адресу. Такой виртуальный адрес называется базовым, и он 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.

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 базовый виртуальный адрес.

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.

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 (с учетом ограничения на максимально возможный адрес, заданного параметром `maxvirtualip`).

Секция [debug]

Секция `[debug]` определяет параметры ведения журнала устранения неполадок управляющего демона (см. «Журналы устранения неполадок ПО ViPNet Coordinator Linux» на стр. 166). Она содержит следующие параметры:

- `debuglevel` – уровень отладки, число от -1 до 5, по умолчанию 3. Значение параметра -1 отключает ведение журнала.
- `debuglogfile` – идентификатор, определяющий место хранения журнала. Формат данного идентификатора следующий: `<спецификатор протокола>:<спецификатор URL для данного протокола>`. Подробное описание возможных значений данного параметра приведено в разделе Журналы устранения неполадок ПО ViPNet Coordinator Linux (на стр. 166). Значение параметра по умолчанию: `file:/var/log/iplircfg.debug.log`, что соответствует записи журнала в указанный файл.

Настройка параметров сетевых интерфейсов

Параметры настройки сетевых интерфейсов содержатся в файлах `iplir.conf-<интерфейс>`. Файлы конфигурации для интерфейсов содержат одну секцию `[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).

  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.

- `timedif` – 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 TCP. Может принимать значение `on` или `off`, значение по умолчанию — `off`.

Как правило, порт клиента при TCP-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 TCP-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. Это часто бывает удобно.

Настройка режимов работы через межсетевой экран

Если ViPNet Coordinator Linux работает во внутренней сети с частными IP-адресами и на границе этой сети установлен межсетевой экран или другое устройство, осуществляющее трансляцию адресов (см. «Трансляция сетевых адресов (NAT)» на стр. 211), то для нормальной работы с другими узлами, находящимися во внешней сети или стоящими за другими межсетевыми экранами, необходимо настроить один из режимов работы через межсетевой экран.

При настройке режима следует помнить, что перед редактированием файла конфигурации `iplir.conf` необходимо остановить управляющий демон, а после окончания редактирования вновь запустить его, чтобы все изменения вступили в силу.

Настройка режима «Без использования межсетевого экрана»

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

1. В секции `[id]` для собственного узла установите параметр `usefirewall` в значение `off`, то есть отключите использование внешнего межсетевого экрана.
2. В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `off`.
3. Для всех используемых сетевых интерфейсов в секциях `[adapter]` установите параметр `type` в значение `internal`.

Внимание! В предыдущих версиях параметр `usefirewall` имел другое назначение, поэтому при возврате к более ранним версиях необходимо снова настроить параметры собственного узла, руководствуясь документацией к устанавливаемой версии. В противном случае будет нарушена логика работы ПО ViPNet Coordinator Linux.

Настройка режима «Координатор»

Если на границе локальной сети установлен координатор, то узлы, имеющие с ним связь, могут выбрать этот координатор в качестве узла, через который и от имени которого будет осуществляться обмен информацией между узлами, находящимися в разных сетях.

При работе узлов через координатор маршрутизируется трафик следующих узлов:

- узлов, которые находятся со стороны других сетевых интерфейсов координатора;

- узлов, которые не работают через этот координатор;
- узлов, которые недоступны по широковещательным пакетам.

Трафик данных узлов автоматически маршрутизируется на адрес координатора (производится подмена IP- и MAC-адреса), который, выполнив подмену адресов, направляет пакеты дальше от своего имени. Аналогичным образом трафик следует в обратную сторону.

Автоматическая маршрутизация зашифрованных пакетов на собственный координатор позволяет не устанавливать на него шлюз по умолчанию. Это позволяет при необходимости маршрутизировать открытые пакеты на другие шлюзы и не делать дополнительных настроек на компьютере после установки ПО ViPNet.

Примечание. Автоматическая подмена MAC-адреса производится только при использовании патча ядра (см. «Использование патча ядра» на стр. 23). В случае использования технологии NETFILTER (см. «Использование технологии NETFILTER» на стр. 25) автоматическая подмена MAC-адресов невозможна, поэтому необходимо использовать шлюз по умолчанию, в качестве которого может использоваться как сам координатор, так и любой другой маршрутизатор в данной сети, который будет перенаправлять пакеты на координатор.

Если для работы через координатор настраивается узел большой локальной сети, разделенной на сегменты различного рода маршрутизаторами и коммутаторами, на которых, как правило, в целях безопасности настраиваются допустимые для пропуска адреса и протоколы, то в этом случае задача администратора значительно упрощается, поскольку не потребуется настройка множества допустимых внешних адресов и протоколов. В этой ситуации будет циркулировать только UDP-трафик с внутренними адресами своего сегмента и внутренним адресом координатора.

Узел может устанавливаться за координатор, только если один из сетевых интерфейсов этого координатора доступен данному узлу по реальному адресу (то есть между ними нет никаких межсетевых экранов).

В качестве координатора, выполняющего функции прокси-сервера, можно выбрать любой из координаторов, доступных данному узлу.

Для настройки работы узла через координатор задайте в файле `iplir.conf` следующие параметры:

1. В секции `[id]`, описывающей координатор, выбранный в качестве прокси-сервера, задайте в параметре `ip` любой из его реальных IP-адресов, доступных данному узлу, если он еще не задан; в этой же секции задайте значение параметра `port`, то есть порта назначения, на который будут посылаться пакеты для данного координатора.
2. В секции `[adapter]`, соответствующей сетевому интерфейсу, со стороны которого находится выбранный координатор, установите параметр `type` в значение `external`.
3. В собственной секции `[id]` установите параметр `usefirewall` в значение `on`; в этой же секции установите параметр `proxyid` равным значению `id` выбранного координатора.
4. В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `off`.

После соединения с координатором и при его правильной настройке в секции `[id]` для выбранного координатора будут установлены значения параметров `firewallip`, `port` и `proxyid`.

Кроме того, могут измениться значения параметров `firewallip` и `port` в секции `[id]` для собственного узла — они должны соответствовать параметрам выбранного координатора.

Настройка режима «Со статической трансляцией адресов»

Если в локальной сети установлен межсетевой экран, выполняющий NAT, на котором можно настроить статические правила NAT, обеспечивающие взаимодействие с определенным внутренним адресом сети по протоколу UDP с заданным портом, и есть необходимость во взаимодействии с другими узлами, находящимися во внешней относительно межсетевого экрана сети, то на узле нужно настроить режим работы со статической трансляцией адресов.

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

На межсетевом экране (или NAT-устройстве) должны быть заданы следующие статические правила NAT:

- Пропускать исходящие UDP-пакеты с IP-адресом и портом данного узла (source-порт) на любой внешний адрес и порт (с подменой адреса источника на внешний адрес NAT-устройства).
- Пропускать и перенаправлять входящие UDP-пакеты с портом данного узла (destination-порт) на IP-адрес данного узла.

Для настройки работы узла через межсетевой экран со статическим NAT задайте в файле `iplir.conf` следующие параметры:

1. В собственной секции `[id]` установите параметр `usefirewall` в значение `on`.
2. В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `off`.
3. В собственной секции `[id]` установите параметр `proxyid` в значение `0`.
4. При необходимости измените значение параметра `port` (по умолчанию — 55777). Этот параметр определяет номер порта, с которого преобразованные в UDP-формат пакеты уходят с данного узла (source-порт) и на который такие пакеты приходят на данный узел (destination-порт). Изменять номер порта нужно в том случае, если внутри локальной сети через один межсетевой экран (или NAT-устройство) работает несколько узлов с ПО ViPNet. У всех таких узлов номера портов должны быть разными.
5. В собственной секции `[id]` установите параметр `fixfirewall` в значение:
   - `on` — если на внешнем межсетевом экране с NAT есть возможность настроить статические правила только для входящих пакетов, предназначенных данному узлу, то есть обеспечить пропуск пакетов, имеющих заданный адрес и порт назначения, а также их перенаправление на адрес данного узла. В этом случае IP-адрес данного узла 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]`.

- `off` — если внешний адрес межсетевого экрана с NAT в процессе работы может меняться. В этом случае на других узлах IP-адрес доступа к данному узлу 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.

6 Если параметру `fixfirewall` было задано значение `on`, то в секции `[adapter]`, соответствующей сетевому интерфейсу, со стороны которого находится внешний межсетевой экран, установите параметр `type` в значение `external`.

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

Настройка режима «С динамической трансляцией адресов»

Если в локальной сети подключение происходит через некоторое устройство, выполняющее трансляцию адресов (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 двустороннее 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, беспроводных устройств, и в других случаях. Затруднительно также произвести настройки на NAT-устройствах, установленных у провайдера (в домашних сетях, 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 двусторонней работы на узле, подключенном к внешней сети 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 (задается параметром `forward_id` в секции `[dynamic]`). Узел в режиме работы через межсетевой экран с динамическим NAT после подключения к сети производит с заданным периодом (по умолчанию — 25 секунд) отправку UDP-пакетов на свой координатор входящих соединений. Период отправки этих пакетов (задается параметром `timeout` в секции `[dynamic]`) не должен намного превышать тайм-аут сохранности динамического правила на NAT-устройстве. Для разных NAT-устройств может быть установлен разный тайм-аут, 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. Такая технология обеспечивает постоянную доступность узла, работающего через NAT-устройство (так как на NAT-устройстве не удаляются динамические правила), и одновременно существенно увеличивает скорость обмена между узлами.

Рисунок 1. 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` в 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, то есть технология, описанная выше, использоваться не будет. В этом режиме из-за удлинения маршрута прохождения пакетов возможно снижение скорости обмена.

Для настройки работы ViPNet Coordinator Linux в данном режиме задайте в файле `iplir.conf` следующие параметры:

1. В собственной секции `[id]` установите параметр `usefirewall` в значение `on`.
2. В собственной секции `[id]` установите параметр `port` в значение из диапазона 1-65535 (обычно 55777), если он еще не задан.
3. В секции `[dynamic]` установите параметр `dynamic_proxy` в значение `on`.

4 В секции `[dynamic]` установите параметр `forward_id` равным значению `id` внешнего координатора для организации входящих соединений. Данный параметр задается вручную, но не может принимать нулевое значение.

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

5 При необходимости измените значения параметров `timeout` и `always_use_server` в секции `[dynamic]`.

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

> **Примечание.** На координаторе входящих соединений узла, настроенного для работы в режиме **С динамической трансляцией адресов**, в секции `[id]` этого узла параметру `proxyid` будет присвоено значение `0xFFFFFFFE`. На всех остальных координаторах в секции `[id]` этого узла в параметр `proxyid` будет записан ViPNet-идентификатор его координатора входящих соединений.

Настройка координатора, выполняющего функции сервера открытого Интернета

Для организации доступа к Интернету с высоким уровнем защиты рекомендуется использовать технологию «Открытый Интернет». Данная технология позволяет произвести отключение компьютера сотрудника от корпоративной сети и предоставляет ему безопасный доступ к ресурсам сети Интернет.

Реализация технологии «Открытый Интернет» состоит в следующем: в сети выделяется виртуальный изолированный сегмент, который будет связан как с координатором с функцией «Открытый Интернет», так и с другими узлами локальной сети. Доступ в Интернет регулирует прокси-сервер прикладного уровня типа Squid, который может находиться на сервере открытого Интернета или на туннелируемом им компьютере.

Внимание! Для обеспечения безопасности вашей сети не рекомендуется размещать прокси-сервер на одном компьютере с сервером открытого Интернета. Более безопасной является конфигурация, в которой прокси-сервер расположен на открытом узле и туннелируется данным координатором.

Рисунок 2. Технология «Открытый Интернет» с выделенным прокси-сервером

Клиенты ViPNet, связанные с сервером открытого Интернета, могут работать только в одном из двух режимов:

- Работа в Интернете. В этом режиме заблокированы соединения со всеми защищенными узлами, кроме координатора, выполняющего роль сервера открытого Интернета.

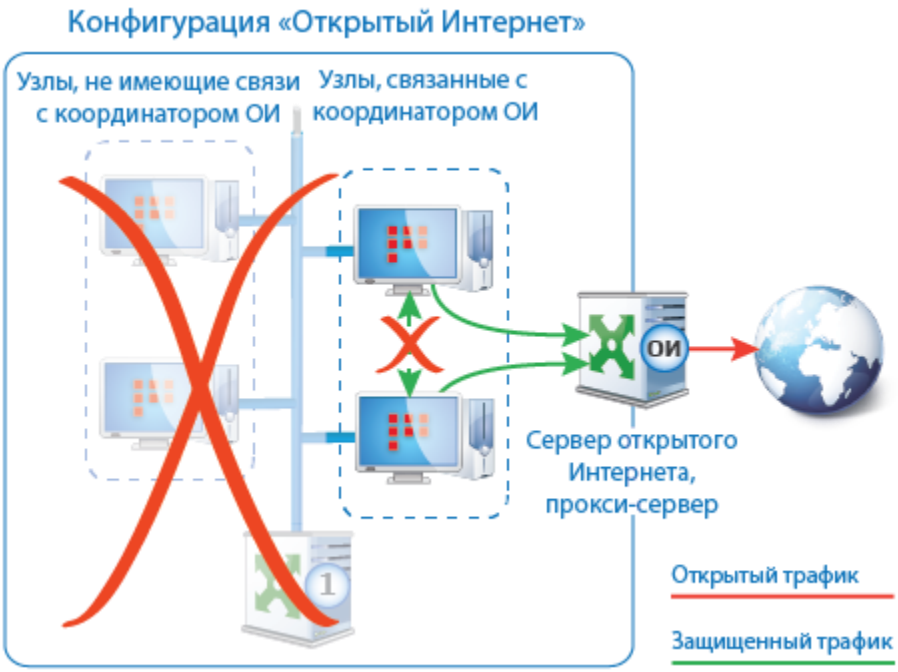

Рисунок 3. Схема работы в режиме «Открытый Интернет»

- Работа с узлами защищенной сети. В этом режиме соединения с сервером открытого Интернета заблокированы, доступ в Интернет невозможен.

Рисунок 4. Схема работы с ресурсами локальной сети

Использование технологии «Открытый Интернет» позволяет решить сразу несколько задач:

- предоставление сотрудникам безопасного и удобного доступа в сеть Интернет с рабочих мест без дополнительных трудозатрат;
- отсутствие затрат на создание и обслуживание специальной выделенной сети, посредством которой сотрудники учреждения получают доступ в сеть Интернет.

Для настройки координатора, выполняющего функции сервера открытого Интернета, необходимо:

- Установить на координатор прокси-сервер прикладного уровня типа Squid.

Примечание. Прокси-сервер может быть установлен на другой компьютер, туннелируемый данным координатором и расположенный за отдельным интерфейсом координатора в демилитаризованной зоне.

- Произвести описанные ниже настройки.

Для настройки координатора, используемого в качестве сервера открытого Интернета, в случае установки прокси-сервера на отдельном компьютере выполните следующие действия в командном интерпретаторе:

1 Добавьте на координаторе правило трансляции адреса прокси-сервера в адрес внешнего интерфейса координатора:

```
firewall nat add src <IP-адрес прокси-сервера> dst @any change src <IP-адрес внешнего интерфейса координатора>
```

2 Добавьте транзитный фильтр открытой сети, разрешающий соединения прокси-сервера с любыми IP-адресами:

```
firewall forward add src <IP-адрес прокси-сервера> dst @any pass
```

3 Добавьте локальный фильтр открытой сети, запрещающий соединения со стороны внутреннего сегмента локальной сети:

```
firewall local add src interface eth0 dst @any drop
```

4 Добавьте фильтр туннелируемых узлов, разрешающий соединения клиентов с прокси-сервером:

```
firewall tunnel add src <диапазон IP-адресов клиентов> dst <IP-адрес прокси-сервера> pass
```

Внимание! Необходимо обратить внимание на настройку маршрутизации на координаторе. Рекомендуется в качестве шлюза по умолчанию указывать шлюз, который сообщит ваш провайдер услуг Интернета. В случае если потребуется обеспечить маршрутизацию между сегментами локальной сети, используйте статические маршруты.

5

Настройка параметров фильтрации IP-трафика

Основные принципы фильтрации трафика 78

Общие сведения о сетевых фильтрах 81

Работа с политиками безопасности 84

Использование групп объектов 85

Компоненты сетевых фильтров 93

Создание сетевого фильтра 99

Просмотр сетевых фильтров 101

Изменение сетевого фильтра 104

Удаление сетевого фильтра 105

Антиспуфинг 107

Дополнительные опции межсетевого экрана 109

Основные принципы фильтрации трафика

Фильтрации подвергается весь трафик, который проходит через сетевой узел:

- открытый (нешифрованный) трафик;
- защищенный (зашифрованный) трафик;
- туннелируемый трафик.

Рисунок 5. Виды IP-трафика

Наибольшую опасность представляет трафик из открытой сети, поскольку в случае атаки достаточно сложно обнаружить ее источник и принять оперативные меры по ее пресечению.

И открытый, и защищенный трафик может быть локальным или широковещательным. Под локальным трафиком понимается входящий или исходящий трафик конкретного узла (то есть когда сетевой узел является отправителем или получателем IP-пакетов). Под широковещательным трафиком имеется в виду передача узлом IP-пакетов, у которых IP-адрес или MAC-адрес назначения является широковещательным адресом (то есть передача пакетов всем узлам определенного сегмента сети).

Кроме этого, через координатор может проходить транзитный трафик. Координатор не является ни отправителем, ни получателем транзитных IP-пакетов, которые следуют через координатор на другие узлы.

Рисунок 6. Виды защищенного и открытого трафика

Для того чтобы правильно настроить сетевые фильтры, необходимо понимать основные принципы фильтрации трафика.

Все входящие и исходящие открытые и защищенные IP-пакеты проходят комплексную фильтрацию в следующей последовательности:

1 Проверка в соответствии с правилами антиспуфинга (см. «Антиспуфинг» на стр. 107).

Примечание. Данная проверка применяется только при фильтрации открытого трафика, в том числе трафика между координатором и его туннелируемыми устройствами.

Если IP-пакет имеет адрес источника, разрешенный правилом антиспуфинга, пакет пропускается. В противном случае — блокируется.

2 Проверка в соответствии с сетевыми фильтрами (см. «Общие сведения о сетевых фильтрах» на стр. 81). Если IP-пакет соответствует параметрам одного из настроенных сетевых фильтров, то он пропускается или блокируется в соответствии с этим фильтром. Если пакет не соответствует ни одному из заданных фильтров, то он блокируется фильтром по умолчанию.

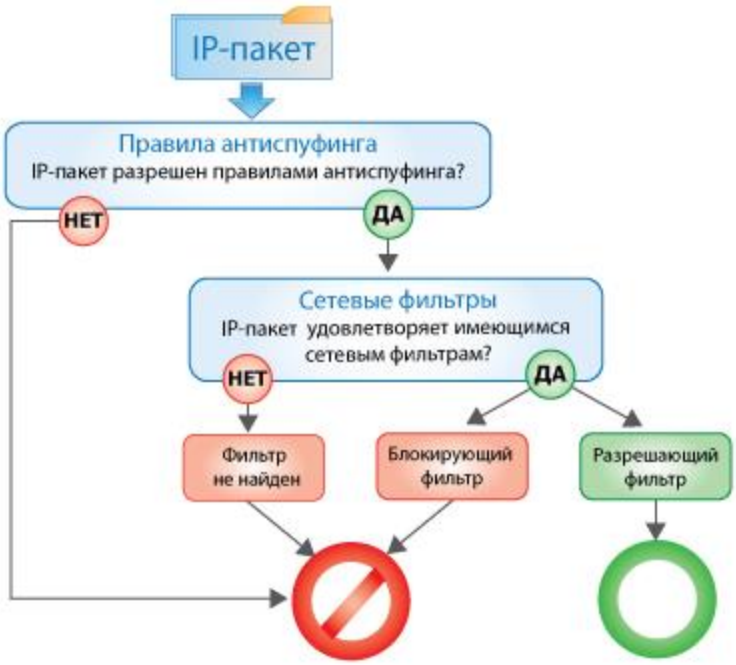

Рисунок 7. Последовательность фильтрации IP-пакетов

Общие сведения о сетевых фильтрах

Сетевые фильтры создаются отдельно для защищенного, открытого (локального и транзитного) и туннелируемого трафика.

С помощью фильтров для открытой сети на защищенном узле можно разрешить либо запретить обмен IP-пакетами с открытыми узлами, то есть с узлами, на которых не установлено программное обеспечение ViPNet с функцией шифрования трафика.

Примечание. К открытым узлам относятся также компьютеры с программным обеспечением ViPNet CryptoService и ViPNet Registration Point.

С помощью фильтров защищенной сети можно ограничить обмен IP-трафиком с защищенными узлами ViPNet, с которыми данный узел имеет связь. Фильтры для туннелируемого трафика определяют правила для IP-пакетов, передаваемых между туннелируемыми узлами и узлами сети ViPNet, с которыми данный координатор имеет связь.

Примечание. Работа с фильтрами для туннелируемых узлов возможна только в случае наличия лицензии на туннелирование хотя бы одного узла.

Сетевые фильтры делятся на три категории:

- Обязательные фильтры. Фильтры данной категории представлены несколькими фильтрами открытой сети:
  - Фильтры, блокирующие открытый входящий и исходящий IP-трафик по TCP- и UDP-протоколам и служебным портам. Передача IP-трафика по указанным протоколам и портам разрешена только сервисам ViPNet.
  - Фильтры, разрешающие открытый IP-трафик, который используется для проверки работоспособности сетевых интерфейсов в режиме работы кластера горячего резервирования.
- Фильтры, поступившие в составе политик безопасности из программы ViPNet Policy Manager.
- Фильтры, заданные в настройках ViPNet Coordinator Linux по умолчанию, и фильтры, добавленные пользователем.

  Если ViPNet Coordinator Linux до версии 4.x обновлялся с версии 3.x, фильтров по умолчанию не будет. Вместо них будут присутствовать фильтры, которые использовались до обновления в сконвертированном формате (см. «Обновление версии ПО ViPNet Coordinator Linux» на стр. 33).

По умолчанию для защищенной сети разрешены только некоторые виды пакетов (служебный трафик ViPNet), поэтому для работы с какими-нибудь дополнительными сервисами в сети ViPNet необходимо настраивать соответствующие сетевые фильтры.

Обязательные фильтры создаются автоматически в ViPNet Coordinator Linux, имеют самый высокий приоритет, то есть применяются в первую очередь, и недоступны для редактирования. После обязательных фильтров размещаются фильтры, поступившие из ViPNet Policy Manager. Эти фильтры также недоступны для редактирования. Самыми последними фильтрами являются фильтры по умолчанию и фильтры, указанные в настройках ViPNet Coordinator Linux. Их можно изменять и удалять.

Последовательность применения сетевых фильтров согласно приоритету изображена на схеме ниже.

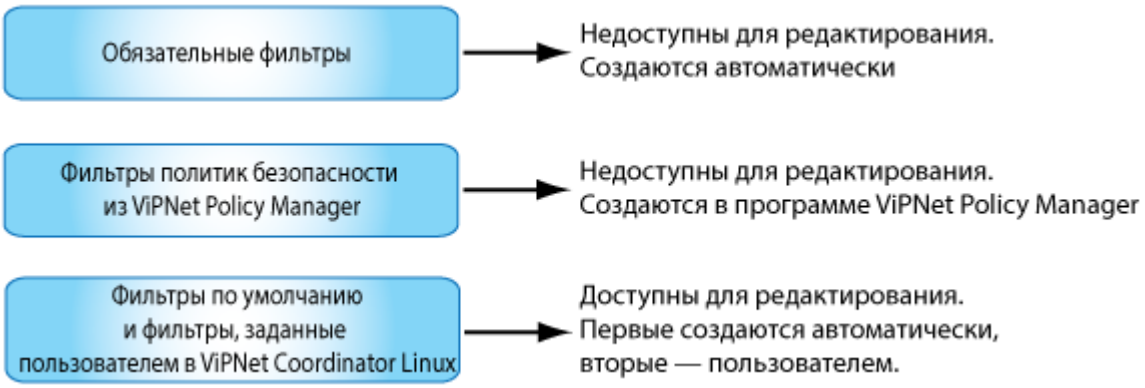

Рисунок 8. Приоритет применения сетевых фильтров

Сетевые фильтры имеют следующие особенности:

- Фильтры включают в себя следующие параметры:
  - Источник и назначение IP-пакетов, на которые распространяется действие фильтра.
  - Протоколы, к которым применяется фильтрация IP-пакетов (TCP, UDP, ICMP и другие).
  - Действие, применяемое к IP-пакетам. Фильтры могут пропускать или блокировать IP-пакеты, соответствующие заданным параметрам.

  Указанные параметры задаются в компонентах сетевых фильтров (см. «Компоненты сетевых фильтров» на стр. 93). Для задания параметров фильтров могут использоваться системные группы объектов (на стр. 86).
- Все IP-пакеты обрабатываются фильтрами по порядку сверху вниз, в соответствии с их положением в таблице. Когда срабатывает первый подходящий под условие фильтр, последующие сетевые фильтры не оказывают на данный пакет никакого влияния.
- Вновь созданные фильтры влияют как на новые, так и на уже существующие соединения. Таким образом, если фильтр, блокирующий трафик соединения, добавлен после установления соединения, то оно будет разорвано.

Настройка сетевых фильтров открытых (нешифрованных) IP-пакетов, а также настройка фильтров защищенной сети производится в командной консоли Linux с помощью командного интерпретатора (см. «Командный интерпретатор ViPNet Coordinator Linux» на стр. 44).

С помощью командного интерпретатора можно настроить следующие сетевые фильтры:

- Локальные фильтры открытой сети. Фильтры IP-пакетов, которыми координатор обменивается с открытыми узлами.
- Транзитные фильтры открытой сети. Фильтры открытых IP-пакетов, проходящих через координатор.
- Фильтры туннелируемых узлов. Фильтры IP-пакетов, передаваемых координатором между туннелируемыми и защищенными узлами.
- Фильтры защищенной сети. Фильтры IP-пакетов, которыми координатор ViPNet обменивается с другими защищенными узлами.

Работа с политиками безопасности

Политика безопасности формируется в программе ViPNet Policy Manager и рассылается на сетевые узлы (координаторы) с помощью транспортного модуля MFTP. Она определяет текущую политику безопасности на узле совместно с сетевыми фильтрами, заданными пользователем на самом узле (в файлах конфигурации).

Политика безопасности, полученная из ViPNet Policy Manager, помещается в тот каталог, где находятся файлы конфигурации.

Политика безопасности включает в себя сетевые фильтры, а также может включать правила трансляции адресов. Сетевые фильтры и правила трансляции, полученные от ViPNet Policy Manager, предшествуют фильтрам и правилам, заданным пользователем на узле, и являются более приоритетными. Обработка политики безопасности на ViPNet Coordinator Linux производится управляющим демоном каждый раз при его старте. Каждый раз при получении политик из ViPNet Policy Manager на узле немедленно применяется новая политика безопасности.

При просмотре сетевые фильтры и правила трансляции, полученные из ViPNet Policy Manager, имеют категорию `Policy` (в столбце `Option`).

Использование групп объектов

Группы объектов — это средство, позволяющее упростить создание сетевых фильтров и правил трансляции в ViPNet Coordinator Linux. Группы объектов объединяют несколько объектов одного типа (например, несколько IP-адресов). При создании фильтров или правил вы можете указать группу вместо перечисления нескольких отдельных объектов.

Группы объектов могут включать следующие типы объектов:

- Узлы ViPNet. Группы узлов ViPNet могут содержать любую комбинацию идентификаторов защищенных узлов, используются при создании фильтров защищенной сети и фильтров туннелируемых узлов.
- IP-адреса. Группы IP-адресов могут содержать любую комбинацию IP-адресов и диапазонов IP-адресов, используются при создании фильтров открытой сети.
- Интерфейсы. Группы интерфейсов содержат любую комбинацию сетевых интерфейсов и используются при создании фильтров открытой сети.
- Протоколы. Группы протоколов содержат любую комбинацию сетевых протоколов и портов, используются в фильтрах открытой и защищенной сети.
- Расписания. Расписания могут содержать любую комбинацию условий выполнения правил (ежедневных, еженедельных или по календарю), используются в фильтрах открытой и защищенной сети.

Группы объектов делятся на несколько видов:

- Системные группы объектов — встроенные в ViPNet Coordinator Linux объекты с фиксированными именами, которые могут использоваться в создаваемых сетевых фильтрах для задания отправителей и получателей IP-пакетов, а также в других пользовательских группах объектов. Системные группы объектов не отображаются в списках групп и их нельзя изменить или удалить. Список системных групп объектов см. в разделе Системные группы объектов (на стр. 86).
- Группы объектов, создаваемые в ПО ViPNet Policy Manager, — группы, которые рассылаются вместе с политиками безопасности. Они недоступны для редактирования и использования в создаваемых сетевых фильтрах, других пользовательских группах объектов.
- Пользовательские группы объектов — группы объектов, создаваемые пользователем непосредственно на узле, а также некоторые группы, настроенные по умолчанию. Подробнее о группах по умолчанию см. в разделе Пользовательские группы объектов, настроенные по умолчанию (на стр. 87). У каждой группы объектов есть свой состав, при этом из состава могут быть заданы некоторые исключения. В состав и исключения группы могут быть включены другие группы объектов той же категории или некоторые системные группы объектов.

Системные группы объектов

Системные группы объектов удобно использовать в сетевых фильтрах и правилах трансляции, так как каждая группа имеет свою область применения и заменяет ввод вручную целого ряда параметров. Например, при необходимости создать сетевой фильтр, блокирующий трафик от всех узлов кроме своего, в фильтре необязательно перечислять адреса всех узлов, а достаточно вставить системную группу объектов `Remote`.

Для добавления системного объекта в сетевой фильтр или правило перед названием системного объекта необходимо вставить символ «`@`» (например, `@Remote`).

Примечание. Системные группы объектов нельзя изменить или удалить.

В таблице ниже приведен список допустимых системных групп объектов и их значений. Также в таблице указано соответствие имен системных групп объектов ViPNet Coordinator Linux и ViPNet Policy Manager.

Таблица 8. Системные группы объектов

Имя группы объектов	Значение	Имя группы объектов в ViPNet Policy Manager
AllClients	Все клиенты из адресного справочника узла.	Все клиенты
AllCoordinators	Все координаторы из адресного справочника узла.	Все координаторы
BroadCast	Все широковещательные адреса. Применяется для идентификации широковещательных адресов только в условии назначения (`dst`) локальных фильтров открытой сети и фильтров защищенной сети. Использование других элементов в условии назначения совместно с системным объектом BroadCast недопустимо.	Широковещательные адреса
Local	Свой узел. Можно указать в качестве источника IP-пакетов для исходящих соединений узла или в качестве назначения для входящих соединений. Использование других элементов в условии источника совместно с системным объектом Local недопустимо.	Мой узел

Имя группы объектов	Значение	Имя группы объектов в ViPNet Policy Manager
Remote	Удаленные узлы (любые узлы, кроме своего). Можно указать в качестве источника IP-пакетов для входящих соединений узла или в качестве назначения для исходящих соединений. Использование других элементов в условии источника совместно с системным объектом Remote недопустимо.	Другие узлы
TunneledIP	Все IP-адреса, туннелируемые узлом (координатором).	Туннелируемые IP-адреса
MultiCast	Диапазон адресов для групповой рассылки (224.0.0.0–239.255.255.255) Можно указать только в качестве назначения для локальных открытых соединений. Использование других элементов в качестве назначения (`dst`) совместно с системным объектом MultiCast недопустимо.	Групповые адреса

Также в сетевых фильтрах и правилах фильтрации может использоваться специализированный объект `@any`, который обозначает любое значение:

- Для локальных и транзитных фильтров открытой сети и правил трансляции — все IP-адреса.
- Для фильтров защищенной сети — все узлы сети ViPNet.
- Для фильтров туннелируемых узлов — все IP-адреса или узлы сети ViPNet.
- Для протоколов — все протоколы.
- Для расписаний — все промежутки времени (круглосуточно).

Пользовательские группы объектов, настроенные по умолчанию

В ПО ViPNet Coordinator Linux имеется ряд предварительно настроенных групп объектов:

- Группы IP-адресов по умолчанию:
  - PrivateNetworkIP (частные IP-адреса) — группа, в составе которой указаны IP-адреса локальных сетей: 10.0.0.0/8; 172.16.0.0/12; 192.168.0.0/16.
  - InternetIP (публичные IP-адреса) — группа, в составе которой указаны все IP-адреса, за исключением частных IP-адресов.

- InternetProxy — группа, в составе которой указан адрес прокси-сервера, поддерживающего протокол HTTP. Данный адрес содержится в переменной окружения Linux: http_proxy.

- Группы расписаний по умолчанию:
  - Workdays (Рабочие дни) — группа с расписанием, в котором заданы рабочие дни недели (понедельник — пятница).
  - Weekends (Выходные дни) — группа с расписанием, в котором заданы выходные дни (суббота и воскресенье).

Также имеется множество предварительно настроенных групп протоколов, которые чаще всего используются при создании сетевых фильтров.

Таблица 9. Группы протоколов, настроенные по умолчанию

Имя группы протоколов	Состав группы
DHCP	UDP:from 67-68 to 67-68
CITRIX	TCP:to 1494
DNS	UDP:to 53
FTP	TCP:to 21
GRE	IP:47
H323	TCP:to 1720
HTTP	TCP:to 80, TCP:to 8080
HTTP-Proxy	TCP:to 3128
HTTPS	TCP:to 443
IGMP	IP:2
IKE	UDP:to 500
IMAP	TCP:to 143
IPSecESP	IP:50
Kerberos	TCP:to 88, TCP:to 749, UDP:to 88, UDP:to 749
L2TP	UDP:to 1701
LDAP	TCP:to 389, UDP:to 389
LotusNotes	TCP:to 1352
MS-SQL	TCP:to 1433-1434, UDP:to 1433-1434
MySQL	TCP:to 3306
NetBIOS-DGM	UDP:from 138 to 138
NetBIOS-NC	UDP:from 137 to 137

Имя группы протоколов	Состав группы
NetMeeting	TCP:to 1503
NTP	UDP:to 123
PING	ICMP:8
POP3	TCP:to 110
Postgres	TCP:to 5432
PPTP	TCP:to 1723
RADIUS	UDP:to 1812-1813
RDP	TCP:to 3389
RTSP	TCP:to 554
SCCP	TCP:to 2000
SIP	TCP:to 5060, UDP:to 5060
SMTP	TCP:to 25
SNMP	UDP:to 161
SNMP-Traps	UDP:to 162
SSH	TCP:to 22
Syslog	UDP:to 514
Telnet	TCP:to 23
TFTP	UDP:to 69
UPnP	TCP:to 1900, TCP:to 2869, UDP:to 1900, UDP:to 2869
MFTP	TCP:to 5000-5003
StateWatcher	TCP:to 2047, TCP:to 5100, TCP:to 10092
ViPNetBase	UDP:to 2046, UDP:from 2048 to 2048, UDP:from 2050 to 2050
Cluster	UDP:from 2060 to 2060
ClusterMonitoring	UDP:from 2060 to 2065, UDP:from 2065 to 2060
SGA	TCP:to 80, TCP:to 5103, TCP:to 10093, TCP:to 10095
WindowsMobileDevices	TCP:to 990, TCP:to 999, TCP:to 5678, TCP:to 5721, TCP:to 26675
WindowsMobileDevices2	UDP:to 5679
VNC	TCP:to 5900

Создание групп объектов

Группы объектов создаются отдельно для каждого типа объектов и сохраняются в таблице соответствующего типа. Поэтому при создании группы объектов обязательно указывается ее тип, название и содержимое.

Чтобы создать группу объектов, выполните следующие действия:

1 В командном интерпретаторе введите `firewall <тип группы> add name <название группы> <содержимое группы> [exclude <исключения группы>]`.

Вы можете создавать группы следующих типов:

- `ip-object` — IP-адреса.
- `vpn-object` — ViPNet-узлы.
- `interface-object` — сетевые интерфейсы.
- `service-object` — протоколы.
- `schedule-object` — расписания.

Название группы начинается с символа «@» и должно быть уникальным в рамках одного типа групп объектов. Объекты, которые будут включены в группу, перечисляются через запятую или указываются в виде диапазона. Подробнее о синтаксисе содержимого групп см. в разделах Условие (на стр. 93) и Расписание (на стр. 96).

Например, чтобы создать группу объектов, включающую сегмент сети за исключением нескольких IP-адресов, выполните следующую команду:

```
firewall ip-object add name @IP_group_1 110.35.14.0/24 exclude
110.35.14.3,110.35.14.13
```

Чтобы создать расписание, по которому фильтр или правило будет применяться только в выходные дни с 9 до 23 часов, выполните следующую команду:

```
firewall shedule-object add name @weekend weekly sa su at 09:00-23:00
```

Примечание. Расписания задаются для установленной на ViPNet Coordinator Linux временной зоны UTC.

2 Нажмите клавишу **Enter**.

Примечание. Если при создании группы объектов возникла ошибка синтаксиса, появится сообщение с указанием ошибки и места, где была допущена опечатка. Повторно введите команду, исправив опечатку, и нажмите клавишу **Enter**.

В результате будет создана группа объектов. Теперь вы можете использовать ее при создании сетевых фильтров (см. «Создание сетевого фильтра» на стр. 99) и правил трансляции (см. «Создание правила трансляции IP-адресов» на стр. 116).

Просмотр групп объектов

С помощью команды `show` вы можете просматривать созданные группы объектов. Для этого в командном интерпретаторе выполните одну из команд:

- `firewall object show` — для просмотра всех групп объектов. Результат выполнения команды будет отображен в виде нескольких таблиц — каждая для отдельного типа групп объектов.
- `firewall <тип группы> show` — для просмотра групп объектов определенного типа. Например, для просмотра всех групп IP-адресов выполните команду:

  ```
  firewall ip-object show
  ```

- `firewall <тип группы> show @<название группы>` — для просмотра конкретной группы объектов. Например, для просмотра группы `@Interface_group` выполните команду:

  ```
  firewall interface-object show @Interface_group
  ```

Пример таблицы при просмотре групп IP-адресов представлен на рисунке ниже.

```
user@ubuntu:~$ vipnet firewall object show
Ip Objects
+====+=================================================+=======================+
|Num |Name                                             |Creation type          |
+----+-----------------------------------+---------------+-----------------------+
|Inclusion                               |Exclusion                              |
+====+===================================+===============+=======================+
|1   |PrivateNetworkIP                                 |User                   |
+----+-----------------------------------+---------------+-----------------------+
|10.0.0.0/255.0.0.0, 172.16.0.0/         |                                       |
|255.240.0.0, 192.168.0.0/255.255.0.0    |                                       |
+====+===================================+===============+=======================+
|2   |InternetIP                                       |User                   |
+----+-----------------------------------+---------------+-----------------------+
|@any                                    |@PrivateNetworkIP                      |
+========================================+=======================================+
```

Рисунок 9. Просмотр групп IP-адресов

Цифрами в таблице обозначены:

1. Заголовок таблицы, где:
   - `Num` — порядковый номер группы объектов.
   - `Name` — название группы объектов.
   - `Creation type` — категория группы объектов (например, для групп, полученных из программы ViPNet Policy Manager, отображается категория Policy).
   - `Inclusion` — содержимое группы объектов.
   - `Exclusion` — исключения группы объектов.
2. Параметры существующих групп объектов.

Удаление групп объектов

С помощью команды `firewall object delete` вы можете удалять группы объектов. Возможно удаление только тех групп, которые не используются в сетевых фильтрах, правилах трансляции и других группах. Если вы хотите удалить группу, которая используется в фильтрах, правилах или других группах, предварительно выполните одно из действий:

- Удалите группу из описания связанных объектов.
- Удалите связанные фильтры, правила или группы.

Чтобы удалить группу объектов, в командном интерпретаторе выполните следующие действия:

1. Введите команду `firewall object delete <название группы>`.
2. Нажмите клавишу **Enter**.
3. В строке подтверждения нажмите клавишу **Y**, чтобы удалить группу или **N**, чтобы отменить удаление.

Примечание. При попытке удаления группы объектов, которая используется в каком-либо сетевом фильтре, правиле трансляции или группе объектов, появится сообщение об ошибке. В сообщении также будет указан список всех зависимых фильтров, правил или групп.

В результате группа объектов будет удалена.

Компоненты сетевых фильтров

Сетевые фильтры любого типа состоят из следующих компонентов:

```
<индекс> rule <название фильтра> <условие> <расписание> <действие>
```

Примечание. Последовательность указанных компонентов в фильтре строго определена, и при ее изменении фильтр невозможно будет сохранить.

Компоненты фильтров, лексемы внутри компонентов, а также служебные слова и параметры внутри лексем отделяются друг от друга пробелами. Лексема представляет собой служебное слово, после которого может указываться какой-либо параметр.

В начале фильтра всегда указываются параметры сетевого фильтра, не относящиеся непосредственно к обработке пакетов:

1. `<индекс>` — указывает порядковый номер фильтра в таблице (`<номер сетевого фильтра>`). Номера используются для обозначения приоритета фильтров — чем меньше номер, тем выше приоритет. При обработке пакета сначала проверяются условия тех фильтров, приоритет которых выше, и при совпадении условий выполняется указанное в фильтре действие, после чего дальнейший просмотр фильтров прекращается. Возможные значения данного параметра — от 1 до последнего номера, использованного в таблице фильтров. При добавлении фильтра, номер которого меньше последнего номера в таблице, нумерация правил, следующих за новым фильтром, будет автоматически изменена (номера будут увеличены на 1).
2. `rule <название фильтра>` — название (описание) сетевого фильтра. Если название фильтра состоит из нескольких слов, разделенных пробелом, необходимо заключать его в кавычки (например, `rule "number one"`).

Примечание. Номер и название сетевых фильтров с помощью параметров `<index>` и `rule <название фильтра>` можно не указывать, поскольку данные параметры являются необязательными. Фильтр без номера будет добавлен в конец таблицы фильтров и будет применяться в последнюю очередь (ему будет присвоен номер, следующий за последним использованным в таблице).

Условие

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

- `src <адрес отправителя>` — описывает условия для адреса отправителя пакета. Можно указать следующие значения:
  - В качестве адреса узла отправителя открытой сети можно указать IP-адрес узла или доменное имя узла.

Примечание. Начиная с версии 4.0, при добавлении в фильтры открытой сети DNS-имени происходит процесс разрешения заданного DNS-имени с помощью отправки соответствующих запросов DNS-серверам. Полученные в ответе DNS-записи сохраняются во внутреннем кэше для дальнейшего использования. Актуальность данных в кэше обеспечивается с помощью отправки дополнительных запросов по истечении времени жизни сохраненных DNS-записей. В случае невозможности успешного разрешения указанного DNS-имени по каким-либо причинам фильтры открытой сети, содержащие такое DNS-имя, не будут загружаться в драйвер до тех пор, пока в процессе отправки повторных запросов не будет получен положительный ответ от DNS-серверов.

Такой подход к использованию DNS-имен в фильтрах открытой сети позволяет значительно сократить время загрузки фильтров в драйвер и обеспечить актуальность используемых DNS-имен.

Если необходимо создать сетевой фильтр для нескольких узлов или для целой сети, можно указать диапазон IP-адресов, доменное имя сети или всю подсеть, указав маску адресов подсети.

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.

Маска адресов — сетевой адрес в формате классовой или бесклассовой адресации CIDR (Classless Internet Domain Routing).

Например: 192.168.1.0/24 или 192.168.1.0/255.255.255.0

Примечание. Если вам требуется задать несколько адресов или доменных имен отправителей в одном условии, то требуется разделять такие элементы запятыми. Например, доменное имя и несколько IP-адресов отправителей в условии фильтра открытой сети должны быть перечислены следующим образом:
`src mydomain.ru,192.168.30.2,192.169.1.1`

- Также при необходимости вместо адреса можно указать ключевые слова `interface` и `byip` для лексемы `src`, где `interface` — параметр, с помощью которого может быть задано системное имя интерфейса, `byip` — параметр, с помощью которого задается IP-адрес интерфейса, диапазон IP-адресов интерфейсов, маска подсети интерфейсов. Параметры `interface` и `byip` являются необязательными и могут быть не указаны. Указание интерфейса может пригодиться, когда вы не можете указать непосредственно IP-адрес узла отправителя, но при этом хотите конкретизировать интерфейс координатора, через который пройдет пакет. Например, при необходимости создать транзитный фильтр открытой сети вы не можете указать IP-адрес своего узла, так как непосредственно ваш узел не является адресом отправителя, но вы можете указать имя или IP-адрес вашего интерфейса, который пересылает пакет по назначению. Сетевой фильтр при этом будет выглядеть следующим образом: `src interface byip 192.168.0.1 dst 192.168.20.2`

Примечание. Если в качестве адреса отправителя пакета указывается доменное имя узла или сети, то после лексемы `src` не следует указывать параметры `interface` или `interface byip`. В случае с доменным именем сетевой фильтр будет выглядеть следующим образом:
`src mydomain.ru`

- В качестве адреса отправителя узла защищенной сети и для фильтров туннелируемых узлов можно указать идентификатор узла (например, 0x1a12000a) или уникальный идентификатор сети (например, 0x1a12).

Внимание! В условиях сетевых фильтров защищенной сети нельзя указывать IP-адреса и DNS-имена узлов, а также другие условия фильтров открытой сети.

- В качестве адреса отправителя также можно указывать группы объектов, в том числе системные группы объектов (на стр. 86). Перед названием группы должен быть указан символ «@». Например, `src @IP_group_1`.

- `dst <адрес получателя>` — описывает условия для адреса получателя пакета. Синтаксис этой лексемы такой же, как синтаксис лексемы `src`, за исключением использования лексем `interface` и `byip`.
- `<протокол>` — задает условия для пропускания или блокировки пакетов, соответствующих какому-либо протоколу. Например, протоколы TCP, UDP, ICMP задаются с помощью лексем `tcp`, `udp`, `icmp` соответственно. Также кроме указанных протоколов можно задать в сетевом фильтре произвольный протокол, указав его номер. Список соответствий между названиями протоколов и их номерами можно посмотреть в файле определений протоколов вашей операционной системы (`/etc/protocols`).

Примечание. Протоколы TCP, UDP и ICMP могут быть заданы как номером, так и названием. В качестве названия используются ключевые слова `tcp`, `udp`, `icmp`, написанные строчными буквами.

Ключевые слова `tcp`, `udp`, `icmp` являются необязательными и могут не указываться. Если требуется задать сразу несколько протоколов, то ключевые слова в сетевом фильтре должны быть указаны через пробел. Если ключевые слова не указаны, то сетевой фильтр будет обрабатывать все типы пакетов независимо от типа протокола, по которому они были переданы.

Для ключевых слов `tcp`, `udp`, `icmp` также могут быть заданы дополнительные параметры:

- Для протоколов TCP, UDP — `sport` и `dport`, где `sport` — порт источника пакета, а `dport` — порт назначения пакета. Параметры `sport` и `dport` могут быть включены в сетевой фильтр как вместе, так и по отдельности. При включении в сетевой фильтр или правило одновременно двух параметров, параметр `sport` всегда должен идти перед параметром `dport`. Также в качестве параметров `sport` и `dport` может быть указан не один порт, а диапазон портов.

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.

Данные параметры являются необязательными и могут не указываться в сетевом фильтре.

- Для протокола ICMP — `type` и `code`, где `type` — тип ICMP-пакета, `code` — код ICMP-пакета. При включении в сетевой фильтр одновременно двух параметров, параметр `type` всегда должен идти перед параметром `code`. Если в условии для протокола ICMP параметр `code` не указан, то считается, что под условие подпадают любые типы ICMP-пакетов.

Внимание! Если протокол задан не ключевым словом (`tcp`, `udp`, `icmp`), а номером, то дополнительные параметры (`sport`, `dport` или `type`, `code`) указать невозможно.

- В качестве протокола для пропускания или блокировки объектов вы можете указать группы протоколов, в том числе пользовательские группы, заданные по умолчанию (см. «Пользовательские группы объектов, настроенные по умолчанию» на стр. 87). Для этого необходимо указать лексему `service` и название группы с символом «`@`», например, `service @FTP`.

Расписание

Расписание позволяет задать временные интервалы, в течение которых действует фильтр. При отсутствии расписания фильтр действует постоянно. Расписание описывается одной из лексем:

- `daily <время>-<время>` — фильтр будет действовать ежедневно в течение указанного промежутка времени. Время указывается в 24-часовом формате `hh:mm`.
- `weekly [mo] [tu] [we] [th] [fr] [sa] [su] [at <время>-<время>]` — фильтр будет действовать еженедельно в указанные дни недели и промежуток времени.
- `calendar <дата>-<дата> [at <время>-<время>]` — фильтр будет действовать в указанные даты и промежуток времени. Дата указывается в формате `DD.MM.YYYY`.
- `schedule <название группы объектов>` — чтобы указать вместо расписания группу объектов. Расписание может быть задано с помощью группы объектов соответствующего типа, если она была создана ранее.

Примечание. Расписания задаются для установленной на ViPNet Coordinator Linux временной зоны UTC.

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 блокирован.

Ограничения на условия сетевых фильтров и правил трансляции

Условия, которые можно применять в качестве источника и назначения сетевых фильтров, зависят от их типа (в какую таблицу добавляется фильтр или правило). При создании фильтров и правил необходимо учитывать следующие ограничения:

- При создании локальных фильтров открытой сети (тип `local`):
  - Вы можете указывать IP-адреса и их диапазоны, DNS-имена, маски адресов, группы IP-адресов, системные имена и адреса сетевых интерфейсов, системные группы `@Local`, `@Remote`, а также `@BroadCast` и `@MultiCast` (только в качестве назначения).
  - Нельзя указывать уникальные идентификаторы защищенных узлов и идентификатор сети ViPNet, группы узлов ViPNet, системные группы `@AllCoordinators`, `@AllClients`.
- При создании транзитных фильтров открытой сети (тип `forward`):
  - Вы можете указывать IP-адреса и их диапазоны, DNS-имена, маски адресов, системные имена и адреса сетевых интерфейсов, группы IP-адресов.
  - Нельзя указывать системные группы объектов, идентификаторы защищенных узлов и сети ViPNet, группы узлов ViPNet.
- При создании фильтров защищенной сети (тип `vpn`):
  - Вы можете использовать идентификаторы узлов и сети ViPNet, группы защищенных узлов, системные группы `@AllCoordinators`, `@AllClients`, `@Local`, `@Remote`, а также `@BroadCast` (только в качестве назначения).
  - Нельзя использовать IP-адреса, DNS-имена, группы IP-адресов, системные имена и адреса сетевых интерфейсов.
- При создании фильтров туннелируемых узлов (тип `tunnel`) нельзя указывать условия для защищенной сети и в качестве источника, и в качестве назначения одновременно. Например, если необходимо обеспечить прохождение трафика от защищенного узла к туннелируемому, то в качестве источника укажите идентификатор этого узла ViPNet, а в качестве назначения — IP-адрес открытого узла. Также вы можете указывать системные группы `@AllClients`, `@AllCoordinators`, `@TunneledIP`.

- При создании правил трансляции IP-адресов (тип `nat`) вы можете указывать только IP-адреса и их диапазоны, DNS-имена, сетевые интерфейсы, а также группы IP-адресов. Для правил трансляции нельзя указывать идентификаторы защищенных узлов и сети ViPNet, группы узлов ViPNet и системные группы объектов.

Создание сетевого фильтра

Сетевые фильтры создаются отдельно для каждого вида трафика и сохраняются в таблице, соответствующей типу фильтра. Поэтому при создании сетевого фильтра обязательно указывается его тип, условие и действие (см. «Компоненты сетевых фильтров» на стр. 93), которое выполняется фильтром.

Примечание. Если в ПО ViPNet Центр управления сетью было запрещено использовать туннелирование на данном координаторе, то при установке ПО ViPNet Coordinator Linux не будут созданы фильтры по умолчанию для туннелируемых узлов. В случае разрешения туннелирования необходимо самостоятельно создать следующий фильтр: `firewall tunnel add src @any dst @any pass`.

Чтобы добавить сетевой фильтр, выполните следующие действия:

1 В командном интерпретаторе введите `firewall <тип сетевого фильтра> add <номер сетевого фильтра> rule <имя сетевого фильтра> <условие> <расписание> <действие>`. Если при создании сетевого фильтра параметр `<номер сетевого фильтра>` управляющего компонента не указывается, фильтр автоматически добавляется в конец таблицы.

   Вы можете задать один из следующих типов сетевых фильтров:

   - `local` – локальные фильтры открытой сети;
   - `forward` – транзитные фильтры открытой сети;
   - `tunnel` – фильтры туннелируемых узлов;
   - `vpn` – фильтры защищенной сети.

   Например, чтобы создать локальный фильтр, блокирующий IP-пакеты, отправляемые с адреса координатора 192.168.30.1 через порт 2525 на открытый узел с адресом 172.16.35.1 на порт 443 по протоколу TCP/IP, выполните команду:

   ```
   firewall local add 1 rule "Rule 1" src 192.168.30.1 dst 172.16.35.1 tcp sport 2525 dport 443 drop
   ```

   Чтобы разрешить FTP-запросы к открытому узлу 192.168.2.4, выполните команду:

   ```
   firewall local add 6 rule "Rule 6" src @local dst 192.168.2.4 service @FTP pass
   ```

   Чтобы создать транзитный фильтр, разрешающий прохождение транзитных IP-пакетов от узла 192.168.0.1 пользователю с адресом 192.168.30.3 через координатор, выполните команду:

   ```
   firewall forward add 2 rule "Rule 2" src 192.168.0.1 dst 192.168.30.3 pass
   ```

   Чтобы создать фильтр, разрешающий отправление IP-пакетов от туннелируемого узла с адресом 192.168.0.1 на защищенный узел с идентификатором 0x1234abab, выполните команду:

   ```
   firewall tunnel add 3 rule "Rule 3" src 192.168.0.1 dst 0x1234abab pass
   ```

   Чтобы создать фильтр, разрешающий отправление IP-пакетов от туннелируемого узла с адресом 192.168.0.1 на туннелируемый узел с адресом 192.168.2.3, выполните команду:

```
firewall tunnel add 4 rule "Rule 4" src 192.168.0.1 dst 192.168.2.3 pass
```

Чтобы создать фильтр защищенной сети, блокирующий пакеты от защищенного узла с идентификатором узла 0x1a12000a до защищенной сети с идентификатором 0x1b14, выполните команду:

```
firewall vpn add rule 5 "Rule 5" src 0x1a12000a dst 0x1b14 drop
```

2 Нажмите клавишу **Enter**.

Примечание. Если при создании сетевого фильтра возникла ошибка, появится сообщение с указанием ошибки синтаксиса и места, где была допущена опечатка. Повторно введите фильтр, исправив опечатку, и нажмите клавишу **Enter**.

В результате будет создан сетевой фильтр. При необходимости вы можете выполнить следующие действия для данного фильтра:

- просмотреть (см. «Просмотр сетевых фильтров» на стр. 101);
- изменить (см. «Изменение сетевого фильтра» на стр. 104);
- удалить (см. «Удаление сетевого фильтра» на стр. 105).

Просмотр сетевых фильтров

С помощью команды `show` можно просмотреть сетевые фильтры, заданные на узле. Описанные ниже команды должны вводиться в командном интерпретаторе.

Чтобы просмотреть сетевые фильтры всех типов, выполните команду:

```
firewall rules show
```

Чтобы просмотреть список сетевых фильтров соответствующего типа, выполните команду:

```
firewall <тип сетевого фильтра> show
```

Например, чтобы просмотреть:

- локальные фильтры открытой сети, заданные на узле, выполните команду:

  ```
  firewall local show
  ```

- транзитные фильтры открытой сети, заданные на узле, выполните команду:

  ```
  firewall forward show
  ```

- фильтры туннелируемых узлов, заданные на узле, выполните команду:

  ```
  firewall tunnel show
  ```

- фильтры защищенной сети, заданные на узле, выполните команду:

  ```
  firewall vpn show
  ```

Чтобы просмотреть конкретный сетевой фильтр, условие или действие которого вы знаете, выполните команду:

```
firewall <тип сетевого фильтра> show <условия поиска>
```

Например, если необходимо просмотреть локальный сетевой фильтр, у которого вы знаете адрес отправителя пакета (например, 192.168.1.10), выполните команду:

```
firewall local show src 192.168.1.10
```

В результате в командном интерпретаторе отобразятся все локальные фильтры, в адресе отправителя которых задан данный IP-адрес.

При задании множественного условия, например, при необходимости задать несколько адресов отправителей в одном условии, требуется разделять такие элементы запятыми, например:

```
firewall local show src mydomain.ru,192.168.30.2,192.169.1.1 dst 192.168.0.1
```

В результате отобразятся все локальные фильтры, в адресе отправителя и получателя которых заданы данные IP-адреса.

В качестве условия поиска могут быть указаны:

- номер сетевого фильтра в таблице;
- название сетевого фильтра;
- адрес отправителя пакета;
- адрес получателя пакета;

- условие для пропускания или блокировки пакетов, соответствующих какому-либо протоколу;
- действие.

Внимание! Поиск сетевых фильтров осуществляется по строгому совпадению с поисковым запросом. Например, под поисковый запрос `src 192.168.1.10` попадут фильтры `src 192.168.1.10` и `src 1.1.1.1,192.168.1.10,2.2.2.2`, но не попадет фильтр `192.168.1.10-192.168.1.11`. Также, например, под запрос `rule Name` попадет `Name`, но не попадет `NameLong`.

Пример таблицы при просмотре локальных фильтров открытой сети приведен ниже.

```
User:
+====+=========================================================+============+==========+
|Num |Name                                                     |Option      |Schedule  |
+----+-------------------------+-----------------------------+------------+----------+
|Act |Source                   |Destination                  |Protocol                 |
+====+=========================+=============================+============+==========+
|1   |Allow DHCP Service                                     |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |udp: from 67 to 68       |
+====+=========================+=============================+============+==========+
|2   |Allow DHCP Service                                     |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |udp: from 68 to 67       |
+====+=========================+=============================+============+==========+
|3   |Allow DHCP-Relay service                               |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |udp: from 67 to 67       |
+====+=========================+=============================+============+==========+
|4   |Allow ICMP Ping                                        |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |icmp: 8                  |
+====+=========================+=============================+============+==========+
|5   |Allow DNS                                              |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@local                   |@any                         |udp: to 53               |
+====+=========================+=============================+============+==========+
|6   |Allow NTP                                              |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@local                   |@any                         |udp: to 123              |
+====+=========================+=============================+============+==========+
|7   |Allow netbios-ns Service                               |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |udp: from 137 to 137     |
+====+=========================+=============================+============+==========+
|8   |netbios-dgm Service                                    |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |udp: from 138 to 138     |
+====+=========================+=============================+============+==========+
|9   |Allow IGMP Traffic                                     |User        |          |
+----+-------------------------+-----------------------------+------------+----------+
|pass|@any                     |@any                         |IP: 2                    |
+====+=========================+=============================+=========================+
```

Рисунок 10. Просмотр локальных фильтров открытой сети

Цифрами в таблице обозначены:

1 Заголовок таблицы, где:

- `Num` — порядковый номер сетевого фильтра.
- `Name` — название сетевого фильтра.

- `Option` — категория сетевого фильтра (например, для фильтров по умолчанию и фильтров, заданных пользователем, отображается категория User).
- `Schedule` — расписание для сетевого фильтра.
- `Act` — действие.
- `Source` — адрес отправителя IP-пакета.
- `Destination` — адрес получателя IP-пакета.
- `Protocol` — протокол, по которому передается IP-пакет.

2 Параметры существующих сетевых фильтров.

Каждый сетевой фильтр условно можно разделить на две части. Первая часть состоит из строки, содержащей номер, наименование и дополнительные (необязательные) параметры сетевого фильтра. Вторая часть состоит из действия, адреса отправителя, адреса получателя и условия фильтра, данная часть может быть представлена несколькими строками.

Изменение сетевого фильтра

Для изменения сетевого фильтра или правила используется команда `change append`.

В сетевой фильтр или правило может быть добавлено:

- условие для адреса отправителя пакета;
- условие для адреса получателя пакета;
- условия для пропускания или блокировки пакетов, соответствующих какому-либо протоколу.

Также можно добавить в сетевой фильтр или правило несколько условий одновременно.

Описанные ниже команды должны вводиться в командном интерпретаторе.

Чтобы добавить условие в существующий фильтр, выполните следующие действия:

1 Введите команду `firewall <тип сетевого фильтра или правила> change append <номер фильтра или правила> <дополнительное условие>`.

   Дополнительное условие будет добавлено к уже имеющемуся условию, которое было задано в фильтре или правиле ранее.

2 Нажмите клавишу **Enter**.

Примечание. Если при сохранении сетевого фильтра или правила возникла ошибка, появится сообщение с указанием синтаксиса ошибки и места, где была допущена опечатка. Повторно введите сетевой фильтр или правило, исправив опечатку, и нажмите клавишу **Enter**.

В результате сетевой фильтр будет изменен.

Например, существует локальный фильтр открытой сети:

```
1 RuleName1 src 192.168.1.1,2.2.2.2/24 dst 10.0.0.1 pass
```

В данный сетевой фильтр необходимо добавить еще одно условие, например, адрес отправителя IP -пакетов. Для этого выполните команду:

```
firewall local change append 1 src 192.168.3.3
```

Далее необходимо подтвердить внесенные изменения.

После добавления адреса отправителя сетевой фильтр будет выглядеть следующим образом:

```
1 RuleName1 src 192.168.1.1,2.2.2.2/24,192.168.3.3 dst 10.0.0.1 pass
```

При необходимости вы можете изменить порядковый номер сетевого фильтра в таблице. Для этого введите команду `firewall <тип сетевого фильтра> move rule <текущий номер фильтра> to <новый номер фильтра>` и нажмите клавишу **Enter**. В результате номер фильтра и, соответственно, его положение в таблице будут изменены.

Удаление сетевого фильтра

Для удаления сетевых фильтров следует использовать команду `delete`. Описанные ниже команды должны вводиться в командном интерпретаторе.

Чтобы удалить фильтр, порядковый номер которого вы знаете:

1. Выполните команду `firewall <тип сетевого фильтра> delete <номер сетевого фильтра>`.
2. Нажмите клавишу **Enter**.
3. В строке подтверждения нажмите клавишу **Y**, если хотите удалить именно этот фильтр, и **N**, если хотите отменить удаление.

В результате сетевой фильтр будет удален.

Например, чтобы удалить локальный сетевой фильтр `1 "Rule 1" src 192.168.1 dst 192.168.4.2 tcp sport 2525 dport 443 pass`:

1. Выполните команду `firewall local delete 1`.
2. Нажмите клавишу **Enter**.
3. Подтвердите удаление, нажав клавишу **Y**.

Если вы не знаете порядковый номер сетевого фильтра, то вы можете сначала найти его по определенным компонентам, а затем удалить его. Для этого:

1. Выполните команду `firewall <тип сетевого фильтра> delete <условия поиска>`

   В качестве условия поиска вы можете указать:

   - Название сетевого фильтра.
   - Адрес отправителя пакета.
   - Адрес получателя пакета.
   - Условие для пропускания или блокировки пакетов, соответствующих какому-либо протоколу.
   - Действие.

   Внимание! Поиск сетевых фильтров осуществляется по строгому совпадению с поисковым запросом. Например, под поисковый запрос `src 192.168.1.10` попадут фильтры `src 192.168.1.10` и `src 1.1.1.1,192.168.1.10,2.2.2.2`, но не попадет фильтр `192.168.1.10-192.168.1.11`. Также, например, под запрос `rule Name` попадет `Name`, но не попадет `NameLong`.

2. Нажмите клавишу **Enter**.

   В результате будут полностью отображаться сетевые фильтры, удовлетворяющие критериям поиска.

3 Из списка найденных сетевых фильтров выберите фильтр, который необходимо удалить, и введите номер удаляемого фильтра. В результате фильтр будет удален.

4 Убедитесь, что сетевой фильтр удален, просмотрев таблицу сетевых фильтров (см. «Просмотр сетевых фильтров» на стр. 101).

Примечание. После удаления сетевого фильтра его порядковый номер присваивается следующему по порядку фильтру в таблице. Последующим фильтрам в таблице также присваиваются следующие по порядку номера.

Например, вы хотите удалить локальный сетевой фильтр, порядковый номер которого вы не знаете, но знаете адрес отправителя пакета, заданный в данном фильтре (например, 192.168.20.2).

Чтобы найти и удалить данный фильтр, выполните команду:

```
firewall local delete src 192.168.20.2
```

В результате будет выведена таблица с локальными фильтрами, подходящими под данное условие. Найдите фильтр в таблице, который необходимо удалить, в поле **Num** узнайте порядковый номер данного фильтра. Введите порядковый номер фильтра и подтвердите его удаление.

Антиспуфинг

Программа ViPNet Coordinator Linux обладает функцией антиспуфинга, то есть блокирования входящих IP-пакетов от отправителей, IP-адреса которых недопустимы на данном сетевом интерфейсе. Антиспуфинг работает только для открытого трафика, поскольку для защищенного трафика IP-адрес отправителя не имеет никакого значения. Открытые пакеты сначала проверяются правилами антиспуфинга, а затем уже обрабатываются сетевыми фильтрами.

Правила антиспуфинга задают для каждого сетевого интерфейса диапазоны IP-адресов, пакеты от которых недопустимы на данном интерфейсе. Пакеты, попадающие в такой диапазон, будут блокироваться.

Как видно из названия, основная задача антиспуфинга — это защита от так называемого спуфинга, одного из видов сетевых атак. При спуфинге злоумышленник посылает какому-либо компьютеру пакет, в котором в качестве адреса отправителя указан не адрес злоумышленника, а какой-либо другой, которому разрешено соединение с данным координатором. Например, таким образом можно отправить открытый пакет из Интернета через координатор, задав в качестве адреса отправителя адрес частной внутренней сети, которая также подключена к данному координатору. Правила антиспуфинга позволяют исключить такую возможность.

Таким образом, для обеспечения высокого уровня безопасности сети рекомендуется, чтобы на координаторе был включен антиспуфинг. По умолчанию антиспуфинг выключен.

Для включения антиспуфинга в командном интерпретаторе введите `iplir option set antispoofing on`.

Для отключения антиспуфинга введите `iplir option set antispoofing off`.

Чтобы просмотреть текущее состояние антиспуфинга, в командном интерпретаторе введите `iplir option get antispoofing`.

Правила антиспуфинга создаются автоматически на основе таблицы маршрутизации сетевого узла. В случае использования сложных схем маршрутизации (с метриками маршрутов или асимметричными маршрутами) функция антиспуфинга может работать некорректно, и ее следует отключить.

Правила антиспуфинга формируются только для открытого транзитного трафика следующим образом:

- Для всех интерфейсов, кроме интерфейса по умолчанию (который соответствует маршруту по умолчанию), блокируются IP-адреса источника, которые не совпадают с адресами, маршрутизируемыми через данный интерфейс.
- Для интерфейса по умолчанию блокируются IP-адреса источника, которые совпадают с зарегистрированными маршрутами других интерфейсов.

Допустим, на сетевом узле используется следующая таблица маршрутизации:

Таблица 10. Пример таблицы маршрутизации

Сетевой адрес	Маска сети	Адрес шлюза	Интерфейс	Метрика
0.0.0.0	0.0.0.0	10.0.8.1	10.0.8.54	20
10.0.8.0	255.255.255.0	On-link	10.0.8.54	276
10.0.8.54	255.255.255.255	On-link	10.0.8.54	276
10.0.8.255	255.255.255.255	On-link	10.0.8.54	276
127.0.0.0	255.0.0.0	On-link	127.0.0.1	306
127.0.0.1	255.255.255.255	On-link	127.0.0.1	306
127.255.255.255	255.255.255.255	On-link	127.0.0.1	306
192.168.48.0	255.255.255.0	On-link	192.168.48.1	276
192.168.48.1	255.255.255.255	On-link	192.168.48.1	276
192.168.48.255	255.255.255.255	On-link	192.168.48.1	276
192.168.59.0	255.255.255.0	On-link	192.168.59.1	276
192.168.59.1	255.255.255.255	On-link	192.168.59.1	276
192.168.59.255	255.255.255.255	On-link	192.168.59.1	276
224.0.0.0	224.0.0.0	On-link	127.0.0.1	306
224.0.0.0	224.0.0.0	On-link	10.0.8.54	276
224.0.0.0	224.0.0.0	On-link	192.168.48.1	276
224.0.0.0	224.0.0.0	On-link	192.168.59.1	276
255.255.255.255	255.255.255.255	On-link	127.0.0.1	306
255.255.255.255	255.255.255.255	On-link	10.0.8.54	276
255.255.255.255	255.255.255.255	On-link	192.168.48.1	276
255.255.255.255	255.255.255.255	On-link	192.168.59.1	276

Рассматриваемый узел имеет четыре сетевых интерфейса. Сетевой интерфейс с адресом 127.0.0.1 является интерфейсом «внутренней петли» (loopback), поэтому не будем его учитывать.

В результате на основе приведенной таблицы маршрутизации будет сформирован следующий набор правил антиспуфинга:

- На сетевом интерфейсе с адресом 192.168.48.1 разрешены входящие пакеты только от IP-адресов 192.168.48.0/24.
- На сетевом интерфейсе с адресом 192.168.59.1 разрешены входящие пакеты только от IP-адреса 192.168.59.0/24.
- На сетевом интерфейсе с адресом 10.0.8.54 разрешены все входящие пакеты, кроме пакетов от IP-адресов 192.168.48.0/24, 192.168.59.0/24.

Дополнительные опции межсетевого экрана

Для возможности тонкой настройки межсетевого экрана в ViPNet Coordinator Linux реализованы следующие дополнительные опции:

- `block-other-protocols` — блокирование пакетов, передающихся не по протоколу IP.

  Драйвер ViPNet анализирует только IP-пакеты и по умолчанию пропускает все пакеты других протоколов, что соответствует установке параметра в значение `off`. Если установить этот параметр в значение `on`, то драйвер 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.

- `max-connections` — максимальное количество одновременных соединений. Значение по умолчанию — 150000. Если необходимо ограничить число одновременных соединений, следует уменьшить значение.

- `dynamic-ports` — диапазон портов, которые используются для динамической трансляции адресов. Значение по умолчанию — 60000-65000.

- `connection-ttl-tcp` — время (тайм-аут) в секундах после регистрации последнего пакета, по истечении которого разрывается данное соединение по протоколу TCP. Значение по умолчанию — 3600 секунд (60 минут).

- `connection-ttl-udp` — время (тайм-аут) в секундах после регистрации последнего пакета, по истечении которого разрывается данное соединение по протоколу UDP. Значение по умолчанию — 300 секунд (5 минут).

- `connection-ttl-ip` — время (тайм-аут) в секундах после регистрации последнего пакета, по истечении которого разрывается данное соединение по протоколу IP. Значение по умолчанию — 60 секунд (1 минута).

- `dynamic-timeouts` — включение или отключение режима динамических тайм-аутов соединений (`on` или `off`). Значение по умолчанию — `off`.

  Режим динамических тайм-аутов используется для противодействия флуд-атакам. Когда количество соединений достигает определенного процента от максимума, тайм-ауты всех соединений уменьшаются на определенную величину. Эта величина тем больше, чем ближе число соединений к максимуму. При этом тайм-ауты не уменьшаются ниже определенного минимума. Когда количество соединений уменьшается до определенного процента от максимального, значения тайм-аутов восстанавливаются до исходной величины.

Управление дополнительными опциями осуществляется с помощью команды `iplir option set <название опции> <значение опции>`.

Просмотреть текущее значение опции можно с помощью команды `iplir option get <название опции>`.

6

Настройка трансляции сетевых адресов (NAT)

Зачем используется трансляция адресов 111

Трансляция адресов в технологии ViPNet 112

Компоненты правил трансляции адресов 115

Создание правила трансляции IP-адресов 116

Просмотр, изменение, удаление правил трансляции адресов 118

Зачем используется трансляция адресов

Трансляция сетевых адресов (NAT, Network Address Translation) — это механизм преобразования IP-адресов одной сети в IP-адреса другой сети. Положения технологии трансляции адресов регламентируются RFC 2663 http://tools.ietf.org/html/rfc2663.

Трансляция сетевых адресов обычно применяется для решения двух основных задач:

- При необходимости подключения локальной сети к Интернету, когда количество узлов локальной сети превышает выданное поставщиком услуг Интернета количество публичных IP-адресов. Таким образом, NAT позволяет локальным сетям, использующим частные адреса, получать доступ к ресурсам Интернета.

  Для решения этой задачи используется трансляция адреса источника (на стр. 113).

- Для организации доступа к внутренним ресурсам из внешней сети. В результате применения технологии NAT локальные сети, имеющие частные адреса, могут быть доступны пользователям Интернета по публичным IP-адресам.

  Для решения этой задачи используется трансляция адреса назначения (на стр. 112).

Правила трансляции адресов могут быть настроены на межсетевом экране — компьютере, разграничивающем локальную (внутреннюю) сеть и глобальную (внешнюю) сеть, например Интернет. Межсетевой экран должен иметь как минимум два сетевых интерфейса:

- Внешний интерфейс — имеет публичный IP-адрес и обеспечивает доступ в Интернет.
- Внутренний интерфейс — имеет частный IP-адрес.

Трансляция сетевых адресов осуществляется для IP-пакетов, проходящих через межсетевой экран из внутренней сети во внешнюю или наоборот.

Трансляция адресов в технологии ViPNet

Внимание! Правила трансляции, описанные в данном разделе, относятся только к открытому трафику. Для защищенного трафика действуют автоматические механизмы трансляции адресов, параметры которых не могут быть изменены.

Узел с программным обеспечением ViPNet Coordinator Linux может выполнять трансляцию адресов (NAT) (см. «Трансляция сетевых адресов (NAT)» на стр. 211) следующих типов:

- Трансляция адреса назначения, называемая также форвардингом портов (port forwarding) 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 координатора, перенаправляются на указанный адрес внутренней сети путем подмены в них адреса получателя, а у ответных пакетов от компьютера внутренней сети подменяется адрес отправителя.
- Трансляция адреса источника, называемая также маскарадингом (masquerading) или динамической трансляцией. Такая трансляция адресов используется, если нужно организовать выход в Интернет пользователей, имеющих частные адреса. В этом случае при проходе через координатор пакетов от частных отправителей в них заменяется адрес отправителя на внешний (реальный) адрес координатора. При приходе ответных пакетов в них подменяется адрес получателя обратно на частный адрес, и в таком виде пакет доставляется в частную сеть.
- Одновременная трансляция адресов источника и назначения. Данная разновидность NAT может использоваться в сложных схемах маршрутизации трафика.

Трансляция сетевых адресов выполняется координатором, только если настроены соответствующие правила (см. «Создание правила трансляции IP-адресов» на стр. 116).

Трансляция адреса назначения

Трансляция адреса узла назначения предназначена для организации доступа из Интернета к серверам локальной сети, не имеющим публичного IP-адреса. Правило трансляции адреса назначения ставит в соответствие частным IP-адресам локальных узлов публичный IP-адрес координатора. В соответствии с правилом, в заголовках IP-пакетов публичный IP-адрес (или IP-адрес и порт) назначения заменяется частным адресом локальной сети. Таким образом, по публичному IP-адресу внешние пользователи могут получить доступ к ресурсам локальной сети.

Рисунок 11. Доступ к внутренним ресурсам при помощи правил трансляции IP-адресов узлов назначения

Если для внешнего IP-адреса координатора задано правило трансляции адреса назначения, то при обращении к этому адресу из Интернета будут выполняться следующие преобразования:

- Во входящих IP-пакетах от внешнего узла координатор подменяет адрес получателя (публичный IP-адрес координатора) локальным адресом в соответствии с описанным правилом. Затем пакет передается через внутренний сетевой интерфейс на узел локальной сети, которому адресован пакет.
- При прохождении ответных пакетов (в рамках уже созданной сессии) координатор производит обратную замену IP-адресов. Адрес отправителя (IP-адрес локального узла) подменяется публичным IP-адресом внешнего сетевого интерфейса координатора. Затем ответный пакет отправляется по назначению (узлу в Интернете).

  Таким образом, при передаче в Интернете пакет выглядит так, будто отправитель и получатель этого пакета имеют публичные IP-адреса.

Внимание! При трансляции адреса узла назначения инициировать соединение может только внешний узел. Чтобы локальный узел мог также иметь доступ в Интернет (двусторонний NAT), необходимо в дополнение к правилу трансляции адреса узла назначения задать также правило трансляции адреса источника.

Трансляция адреса источника

Трансляция адреса источника предназначена для организации доступа локальных компьютеров в Интернет. Правило трансляции адреса источника ставит в соответствие нескольким частным IP-адресам локальных узлов публичный IP-адрес координатора. В соответствии с правилом, в заголовках IP-пакетов частные IP-адреса источника заменяются на публичный IP-адрес. Таким образом, узлы локальной сети могут устанавливать соединения с узлами в Интернете от имени публичного IP-адреса координатора.

Рисунок 12. Организация доступа в Интернет при помощи правила трансляции IP-адреса источника

Если на координаторе настроено правило трансляции адреса источника, то транзитные IP-пакеты, проходящие через координатор из локальной сети в Интернет (или другие глобальные сети) будут преобразованы следующим образом:

- В момент передачи IP-пакета из локальной сети в Интернет ViPNet Coordinator преобразует адрес и (или) порт отправителя пакета для протоколов TCP и UDP. Для пакетов протокола ICMP преобразуется адрес отправителя, остальные параметры запоминаются. В процессе преобразования частный адрес отправителя пакета заменяется на публичный адрес внешнего сетевого интерфейса координатора, обеспечивающего доступ в глобальную сеть. При дальнейшей передаче в Интернете пакет имеет публичный IP-адрес отправителя. Номера портов отправителя (для протоколов TCP и UDP) и запоминаемые параметры (для протокола ICMP) пакетов имеют уникальные значения для всех исходящих IP-соединений внешнего сетевого интерфейса координатора. После преобразования пакет отправляется адресату в Интернете.
- При прохождении ответных пакетов ViPNet Coordinator производит обратное преобразование указанных параметров. То есть в момент передачи ответного IP-пакета ViPNet Coordinator заменяет в нем адрес получателя на частный адрес узла локальной сети, которому адресован ответный пакет. Преобразование происходит на основании уникальных номеров портов, присвоенных исходящим пакетам (для протоколов TCP и UDP), и запоминаемых параметров исходящих пакетов (для протокола ICMP). Номера портов (для протоколов TCP и UDP) также преобразуются в свои истинные значения. Затем ответные пакеты передаются через внутренний сетевой интерфейс узлу локальной сети, которому адресован пакет.

Примечание. Для всех протоколов, кроме TCP, UDP и ICMP, преобразуются только IP-адреса. Для протоколов с частичным преобразованием трансляция IP-адреса источника не будет работать, если несколько узлов локальной сети одновременно инициируют соединение с одним и тем же IP-адресом публичной сети.

Компоненты правил трансляции адресов

Правила трансляции адресов имеют компоненты, подобные компонентам сетевых фильтров (см. «Компоненты сетевых фильтров» на стр. 93). Каждое правило состоит из следующих компонентов:

```
<индекс> rule <название правила> <условие> <действие>
```

Примечание. Последовательность указанных компонентов в правиле строго определена и при ее изменении правило невозможно будет сохранить.

В правилах трансляции адресов:

- **Условие** для правил трансляции адресов имеет такой же синтаксис, как и для сетевых фильтров, но с некоторыми особенностями. Для трансляции адреса отправителя и адресов получателя в лексемах `src` и `dst` можно указывать только адреса, диапазоны адресов и маски, доменные имена и системные группы объектов (на стр. 86). Указывать идентификаторы ViPNet-узлов или ViPNet-сети нельзя.
- **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 источника: `change src <адрес>`, где `<адрес>`:
    - Внешний IP-адрес координатора, на который будет заменяться адрес отправителя пакетов. Например: `change src 194.87.0.8`.
    - Параметр `auto`. В этом случае в качестве адреса, на который будет заменяться адрес отправителя, автоматически будет подставляться адрес интерфейса координатора, с которого будет отправляться пакет.
  - Для трансляции адреса назначения: `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`. Параметр `<порт>` является необязательным и может не указываться.

Создание правила трансляции IP-адресов

Описанные в данном разделе команды должны выполняться в командном интерпретаторе.

Примечание. Правила трансляции IP-адресов работают только при наличии соответствующих транзитных фильтров открытой сети (см. «Создание сетевого фильтра» на стр. 99). Сетевые фильтры должны пропускать трафик определенных узлов, к которому будет применяться трансляция адресов.

Чтобы создать правило трансляции адресов:

1 Выполните команду `firewall nat add <управляющий компонент> <условие> <действие>`.

Например, если при отправлении IP-пакета с адреса 10.0.0.1 на внешний адрес узла 192.168.20.1 необходимо частный адрес отправителя пакета заменять на публичный адрес внешнего сетевого интерфейса координатора, то нужно создать правило трансляции адреса источника. Для этого выполните команду:

```
firewall nat add rule "Rule 1" src 10.0.0.1 dst 192.168.20.1 change src auto
```

Если при отправлении IP-пакета от внешнего узла с адресом mydomain.ru на адрес узла 192.168.20.1 необходимо, чтобы координатор подменял адрес получателя (публичный IP-адрес координатора) на локальный адрес, то нужно создать правило трансляции адреса назначения. Для этого выполните команду:

```
firewall nat add rule "Rule 2" src mydomain.ru dst 192.168.20.1 change dst
10.0.0.7
```

Если при отправлении IP-пакета от внешнего узла с адресом mydomain.ru на адрес узла 192.168.20.1 по протоколу TCP/IP через порт 8080 необходимо, чтобы координатор подменял адрес получателя (публичный IP-адрес координатора) на локальный адрес, то нужно создать правило трансляции адреса назначения. Для этого выполните команду:

```
firewall nat add rule "Rule 3" src mydomain.ru dst 192.168.20.1 tcp dport 8080
change dst 10.0.0.7:8080
```

Если необходимо одновременно транслировать адреса источника и назначения, например, от адреса 10.0.2.15 до адреса 192.168.1.2, выполните команду:

```
firewall nat add src 10.0.2.15 dst 192.168.1.2 change src auto dst 10.0.2.15
```

2 Нажмите клавишу **Enter**.

Примечание. Если при создании правила трансляции возникла ошибка, появится сообщение с указанием синтаксиса ошибки и места, где была допущена опечатка. Повторно введите правило, исправив опечатку, и нажмите клавишу **Enter**.

В результате будет создано правило трансляции IP-адресов. При необходимости вы можете просмотреть, изменить данное правило или удалить его (см. «Просмотр, изменение, удаление правил трансляции адресов» на стр. 118).

Просмотр, изменение, удаление правил трансляции адресов

Просмотр правил трансляции адресов осуществляется аналогично просмотру сетевых фильтров (см. «Просмотр сетевых фильтров» на стр. 101). Чтобы просмотреть список правил трансляции IP-адресов, заданных на узле, выполните команду:

```
firewall nat show
```

Поиск правил трансляции также, как и сетевых фильтров, может осуществляться по номеру, названию правила, адресу отправителя или получателя IP-пакета и другим компонентам, а также по компонентам `change src` (трансляция адреса отправителя) и `change dst` (трансляция адреса получателя).

Пример таблицы при просмотре правил трансляции IP-адресов приведен ниже.

```
User:
+====+=================================================+=======================+
|Num |Name                                             |Option                 |
+----+-------------------------+-----------------------+-----------------------+
|Act |Source                   |Destination            |Protocol               |
+====+=========================+=======================+=======================+
|1   |                                                 |User                   |
+----+-------------------------+-----------------------+-----------------------+
|NAT |@any                     |89.175.26.2            |tcp: to 80             |
|    |                         |Change: 192.168.1.2    |Change DstPort: 25     |
+====+=========================+=======================+=======================+
|2   |                                                 |User                   |
+----+-------------------------+-----------------------+-----------------------+
|NAT |@any                     |89.175.26.3            |tcp: to 81             |
|    |                         |Change: 192.168.1.3    |Change DstPort: 80     |
+====+=========================+=======================+=======================+
|3   |                                                 |User                   |
+----+-------------------------+-----------------------+-----------------------+
|NAT |@any                     |89.175.26.4            |tcp: to 82             |
|    |                         |Change: 192.168.1.3    |Change DstPort: 21     |
+====+=========================+=======================+=======================+
```

Рисунок 13. Просмотр правил трансляции адресов

Для правил трансляции полю `Act` присваивается значение `NAT`, а также в полях `Source` и `Destination` отображаются IP-адреса до и после трансляции, в зависимости от трансляции адреса получателя или адреса отправителя соответственно.

Изменение и удаление правил трансляции адресов осуществляется аналогично изменению (см. «Изменение сетевого фильтра» на стр. 104) и удалению сетевых фильтров (см. «Удаление сетевого фильтра» на стр. 105).

7

Настройка параметров обработки прикладных протоколов

Общие сведения о прикладных протоколах 120

Описание прикладных протоколов 122

Настройка параметров обработки прикладных протоколов 123

Общие сведения о прикладных протоколах

Функционирование сетевых сервисов (например, IP-телефонии, DNS-службы, FTP-службы) обеспечивается прикладными протоколами. При использовании прикладных протоколов IP-адреса часто передаются в теле IP-пакета. Поведение подобного рода может привести к отсутствию сервиса на защищаемых ресурсах в случае использования технологии виртуальных IP-адресов или трансляции адресов. Кроме того, некоторые протоколы, помимо основного (управляющего) соединения, открывают для передачи данных дополнительные соединения на случайно выбранный порт. Для IP-пакетов, следующих на порт назначения, номер которого заранее не известен, невозможно создать разрешающий фильтр, следовательно, соединение будет заблокировано.

Решить перечисленные проблемы позволяет функция обработки прикладных протоколов, которая обеспечивает:

- Подмену виртуального IP-адреса в теле пакета на реальный IP-адрес в случае использования технологии виртуальных IP-адресов.
- Подмену IP-адреса защищенного узла в прикладном протоколе на транслируемый адрес в случае использования технологии трансляции адресов.
- Активацию разрешающего сетевого фильтра для дополнительного соединения на случайно выбранный порт, открываемый прикладным протоколом.

Примечание. Обработка прикладных протоколов осуществляется для всех видов трафика: открытого, защищенного и туннелируемого.

Следует учитывать, что обработка прикладных протоколов не предполагает автоматического разрешения на установление управляющего соединения с открытыми узлами. Установление управляющего соединения с открытыми узлами осуществляется в соответствии с настроенными фильтрами трафика (см. «Общие сведения о сетевых фильтрах» на стр. 81).

Рассмотрим обработку прикладного протокола на примере протокола FTP.

При передаче файлов между FTP-клиентом и FTP-сервером протокол регламентирует установление двух TCP-соединений: управляющее соединение (для отправки команд FTP-серверу и получения ответов от него) и дополнительное соединение (для передачи данных). Соединение клиента с сервером осуществляется в одном из двух режимов: активном и пассивном. В активном режиме клиент инициирует управляющее соединение с порта из диапазона 1024–65535 на порт с номером 21 на сервере. По номеру порта, с которого клиент инициировал соединение, сервер подключается к клиенту и устанавливает соединение для передачи данных. При этом со стороны сервера соединение происходит через порт с номером 20. В пассивном режиме после установления управляющего соединения сервер сообщает клиенту случайно выбранный номер порта из диапазона 1024–65535, к которому можно подключиться при установлении соединения

для передачи данных. Таким образом, в активном режиме клиент должен принять соединение для передачи данных от сервера, в пассивном режиме соединение для передачи данных всегда инициирует клиент.

Для установления управляющего и дополнительного соединений в активном или пассивном режиме работы протокола FTP выполните следующие настройки:

- Создайте фильтр открытой сети, разрешающий исходящее соединение по протоколу TCP на порт 21 FTP-сервера.
- Для разрешения дополнительного соединения в активном режиме работы должна быть включена обработка протокола FTP, которая активирует необходимый фильтр трафика. Убедитесь, что она включена.
- Для разрешения дополнительного соединения в пассивном режиме работы специальные настройки не требуются.

Рассмотрим еще один пример — обработку прикладного протокола на примере протокола SIP.

Протокол SIP предназначен для организации, модификации и завершения сеансов связи — мультимедийных конференций, телефонных соединений — и распределения мультимедийной информации.

Вызывающий SIP-клиент отправляет запрос (например, приглашение к сеансу связи, подтверждение приема ответа на запрос, завершение сеанса связи) вызываемому SIP-клиенту с указанием его SIP-адреса. В зависимости от способа установления соединения запрос направляется вызываемому клиенту либо напрямую, либо с участием прокси-сервера SIP, либо с участием сервера переадресации. Вызываемый клиент в зависимости от типа полученного запроса передает вызывающему клиенту ответ на запрос (например, информацию об ошибке при обработке запроса, успешной обработке запроса, отклонении входящего вызова).

Для установления сеанса связи между SIP-клиентами протокол SIP регламентирует установление соединений TCP и UDP через порт 5060.

Чтобы установить сеанс связи между SIP-клиентами, убедитесь, что включена обработка протокола SIP, и создайте фильтр открытой сети, разрешающий входящее и исходящее соединение по протоколам TCP и UDP на порт 5060 (если хотя бы один из SIP-клиентов является открытым узлом).

Описание прикладных протоколов

В ViPNet Coordinator Linux реализована возможность настройки параметров обработки следующих прикладных протоколов:

- Протокол FTP обеспечивает передачу файлов между FTP-клиентом и FTP-сервером.
- Протокол DNS (Domain Name System) обеспечивает разрешение DNS-имен сетевых узлов в IP-адреса.
- Протокол H.323 обеспечивает работу программ для проведения мультимедийных конференций через IP-сети, в том числе Интернет.
- Протокол SCCP (Skinny Client Control Protocol) обеспечивает передачу сообщений между Skinny-клиентами (проводными и беспроводными IP-телефонами Cisco) и сервером голосовой почты Cisco Unity и Cisco CallManager.
- Протокол SIP (Session Initiation Protocol) обеспечивает установление сеансов связи при передаче голосовых звонков, видеоконференций, а также мультимедийной информации.

Настройка параметров обработки прикладных протоколов

В ViPNet Coordinator Linux параметры обработки прикладных протоколов заданы по умолчанию и соответствуют значениям, указанным в таблице.

Таблица 11. Настройки обработки прикладных протоколов по умолчанию

Прикладной протокол	Сетевой протокол и порт для обработки
FTP	TCP, 21
DNS	UDP, 53
H.323	TCP, 1720
SCCP	TCP, 2000
SIP	TCP, 5060; UDP, 5060

Примечание. По умолчанию для всех прикладных протоколов заданы наиболее часто используемые сетевые протоколы и порты.

Список поддерживаемых прикладных протоколов задан по умолчанию, нельзя добавить протоколы или удалить протоколы из списка.

Вы можете изменить настройки параметров обработки прикладных протоколов, используя команды группы `alg` (Application Layer Gateway). С помощью команд данной группы вы можете:

- просмотреть текущие настройки параметров обработки прикладных протоколов;
- изменить настройки обработки прикладных протоколов;
- включить или выключить обработку прикладных протоколов;
- запустить или остановить демон обработки прикладных протоколов algd.

Примечание. При перезапуске демона algd (с помощью команд `alg stop`, `alg start`) в случае, если демон iplircfg запущен, установленные до перезапуска соединения могут продолжать функционировать до 5 минут. Новые соединения для данных узлов в это время будут недоступны.

Для обеспечения бесперебойной работы прикладных сервисов в процессе штатного функционирования ПО ViPNet Coordinator Linux не рекомендуется останавливать демон algd.

Чтобы просмотреть текущие параметры обработки прикладных протоколов для открытого и защищенного трафика, выполните команду:

```
alg show
```

В результате отобразится таблица с текущими параметрами прикладных протоколов:

```
-> vipnet  alg show
+==============+========+==================================================+=======+
|SERVICE       |PROTOCOL|PORTS                                             |ON/OFF|
+==============+========+==================================================+=======+
|FTP           |TCP     |21                                                |ON    |
+==============+========+==================================================+=======+
|FTP           |UDP     |                                                  |OFF   |
+==============+========+==================================================+=======+
|DNS           |TCP     |                                                  |OFF   |
+==============+========+==================================================+=======+
|DNS           |UDP     |53                                                |ON    |
+==============+========+==================================================+=======+
|H323          |TCP     |1720                                              |ON    |
+==============+========+==================================================+=======+
|H323          |UDP     |                                                  |OFF   |
+==============+========+==================================================+=======+
|SCCP          |TCP     |2000                                              |ON    |
+==============+========+==================================================+=======+
|SCCP          |UDP     |                                                  |OFF   |
+==============+========+==================================================+=======+
|SIP           |TCP     |5060                                              |ON    |
+==============+========+==================================================+=======+
|SIP           |UDP     |5060                                              |ON    |
+==============+========+==================================================+=======+
```

Рисунок 14. Просмотр параметров прикладных протоколов

Чтобы включить обработку прикладных протоколов или изменить настройки обработки прикладных протоколов, в командном интерпретаторе выполните команду:

`alg module <ModuleName> process <Transport> <Ports> on` — для настройки и включения обработки прикладного протокола, где:

- `<modulename>` — имя прикладного протокола (sip / h323 / ftp / sccp / dns);
- `<transport>` — название сетевого протокола для обработки (tcp / udp);

> **Примечание.** Для некоторых прикладных протоколов возможно отсутствие поддержки какого-либо сетевого протокола. Например, не поддерживается сетевой протокол UDP для прикладного протокола FTP.

- `<ports>` — порт или диапазон портов для обработки.

> **Примечание.** Заданные параметры обработки прикладных протоколов должны соответствовать параметрам, указанным в настройках различных приложений, таких как FTP-клиент, DNS-клиент, SIP-клиент и других.

В результате обработка прикладного протокола будет включена и настроена с заданными параметрами.

Рассмотрим процесс изменения параметров обработки прикладного протокола на примере протокола SIP. Например, вам недостаточно использовать стандартные порты для обработки протокола SIP и требуется добавить к стандартным портам свой список портов (например, для UDP требуется добавить порт 10000 и диапазон портов 21-65, а для TCP оставить параметры по умолчанию). Для этого выполните команду:

```
alg module SIP process UDP 5060,10000,21-65 on
```

В результате параметры обработки протокола SIP будут изменены.

Примечание. Для изменения значения портов правило обработки прикладного протокола необходимо создавать заново с нужными параметрами.

Для отключения обработки прикладных протоколов используется команда

```
alg module <ModuleName> process off
```

Чтобы отключить обработку прикладного протокола только на одном транспортном протоколе, установите значение порта 0 для данного транспортного протокола. Например:

```
alg module SIP process TCP 0 on
```

Внимание! Не рекомендуется отключать обработку прикладных протоколов, в противном случае работа прикладных программ может быть затруднена.

8

Система защиты от сбоев

Назначение системы защиты от сбоев 127

Состав системы защиты от сбоев и принципы ее работы 128

Управление системой защиты от сбоев 130

Настройка системы защиты от сбоев 131

Назначение системы защиты от сбоев

Система защиты от сбоев предназначена для создания отказоустойчивого решения на базе ПО ViPNet Coordinator Linux. Данная система имеет два режима функционирования:

1. Одиночный режим (режим одиночного координатора).
2. Режим кластера (режим кластера горячего резервирования серверов).

При работе в одиночном режиме, который устанавливается автоматически при инсталляции ПО ViPNet Coordinator Linux, 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 iplir.

Режим кластера горячего резервирования обеспечивает передачу функций вышедшего из строя сервера другому (резервному) серверу. При работе в режиме кластера система защиты от сбоев также выполняет функции одиночного режима, то есть обеспечивает постоянную работоспособность основных служб, входящих в состав ПО. Подробное описание режима кластера горячего резервирования приведено в документе «ViPNet Coordinator Linux. 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 watchdog и программы-демона failoverd, 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. 131) 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 загрузке 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);
- демон обработки прикладных протоколов (algd);
- сервер веб-интерфейса (axis2.cgi).

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, то есть в течение 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. 131), 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 приложение, так и демон iplircfg 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 отрабатывает 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 так далее. 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`, `mftp stop`, `vipnet-web-gui stop` или `alg stop`), то оно производит дерегистрацию в системе защиты от сбоев, слежение за ним отключается. В этом случае для дальнейшей работы администратор должен вручную запустить приложение (соответственно командой `iplir start`, `mftp start`, `vipnet-web-gui start` или `alg 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 управляющего скрипта failover, который помещается при установке в каталог `/sbin`.

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 ПО ViPNet Coordinator Linux (iplircfg, mftpd, algd, axis2.cgi).

Примечание. При выполнении команды `failover start` вручную в одиночном режиме перезапуск контролируемых приложений не производится.

Запрос информации о текущем состоянии системы защиты от сбоев производится командой `failover info` (см. «Программа просмотра информации о состоянии системы защиты от сбоев» на стр. 142).

Настройка системы защиты от сбоев

Администратор может настраивать параметры работы системы защиты от сбоев путем редактирования файла `failover.ini`, расположенного в каталоге `/etc`.

Внимание! В файле `failover.ini` в соответствующих секциях должны быть указаны все интерфейсы, которые присутствуют в файле `iplir.conf` в секции `[adapter]`. Если данное условие не будет выполнено, возникнет ошибка резервирования.

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. 128). Может принимать значение `yes` или `no`. Значение `yes` включает механизм перезагрузки системы, `no` — выключает. По умолчанию значение параметра `no`.

    При установке предыдущих версий ПО ViPNet Coordinator Linux (до 2.8.18) параметр `reboot` устанавливается автоматически в зависимости от значения параметра `usewatchdog` секции `[watchdog]` (данная секция является устаревшей и удаляется из файла конфигурации при установке текущей версии ПО ViPNet Coordinator Linux). Если параметр `watchdog` был включен (имел значение `on`), то в качестве значения параметра `reboot` устанавливается `yes`, в противном случае – `no`.
- Секция `[debug]` определяет параметры ведения журнала устранения неполадок демона failoverd. Она содержит следующие параметры:
  - `debuglevel` – уровень протоколирования, число от -1 до 5. Значение параметра `-1` отключает ведение журнала.

    Значение по умолчанию — `3`.
  - `debuglogfile` – идентификатор, определяющий место хранения журнала. Формат данного идентификатора следующий: `<спецификатор протокола>:<спецификатор URL для данного протокола>`. Подробное описание возможных значений данного параметра приведено в разделе Журналы устранения неполадок ПО ViPNet Coordinator Linux (на стр.

166). Значение параметра по умолчанию — `file:/var/log/failover.debug.log`, что соответствует записи журнала в указанный файл.

Перед редактированием файла конфигурации `failover.ini` нужно остановить демон failoverd командой `failover stop`, а после редактирования запустить его снова командой `failover start`.

9

Консольные утилиты

Программа просмотра журнала регистрации IP-пакетов 134

Программа просмотра информации о защищенном узле 140

Программа просмотра информации о состоянии системы защиты от сбоев 142

Программа смены пароля пользователя 144

Программа распаковки дистрибутива ключей 145

Программа работы с конфигурациями ViPNet 146

Программа просмотра журнала регистрации IP-пакетов

Программа `dbviewer` предназначена для просмотра журнала регистрации IP-пакетов, который ведется управляющим демоном. При работе с программой просмотра предоставляется возможность вывести записи журнала, 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.

Программа `dbviewer` после установки помещается в каталог `/sbin`. Она запускается из управляющего скрипта `iplir` с помощью команды `iplir view`. После запуска программы появится окно для задания параметров поиска в журнале:

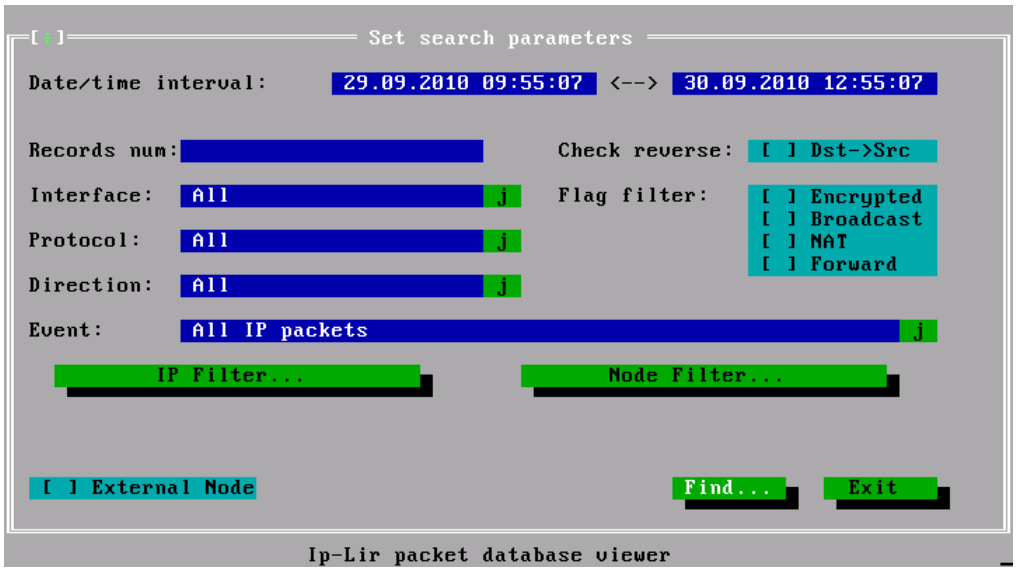

Рисунок 15. Задание параметров поиска в журнале регистрации IP-пакетов

Примечание. Если символы отображаются некорректно, загрузите экранный шрифт, используемый по умолчанию, с помощью одной из команд:

- `setfont`
- `setfont default8x16`
- `consolechars`
- `consolechars -f /usr/share/consolefonts/default8x16.psf.gz`

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 подсоединения к удаленному узлу необходимо, чтобы узел был доступен на уровне TCP/IP и на нем работала программа управления. Если соединиться не удается, программа просмотра журнала выдает сообщение 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:
  - **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**.

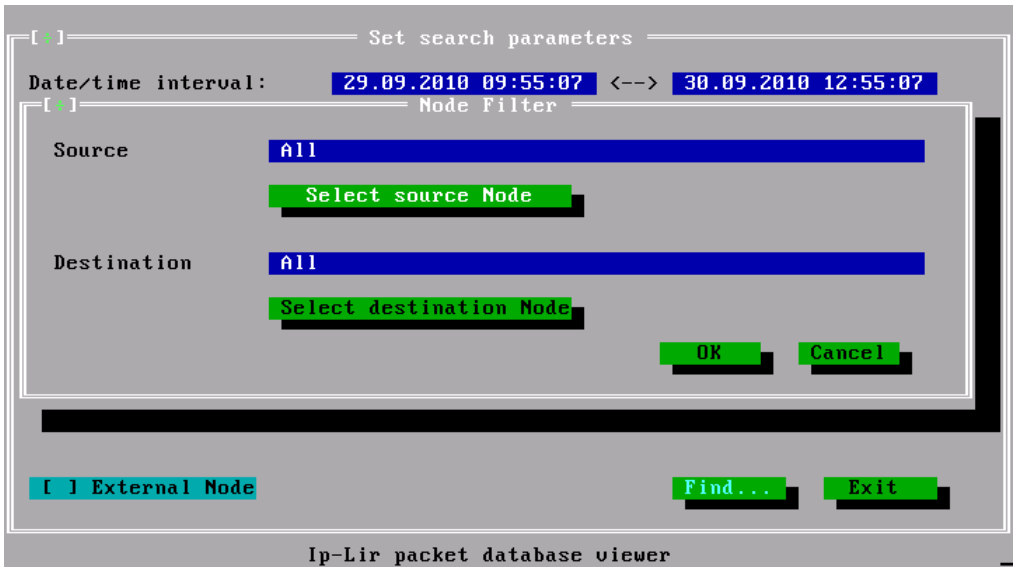

Рисунок 16. Задание параметров запроса по узлу отправителя и/или получателя

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.

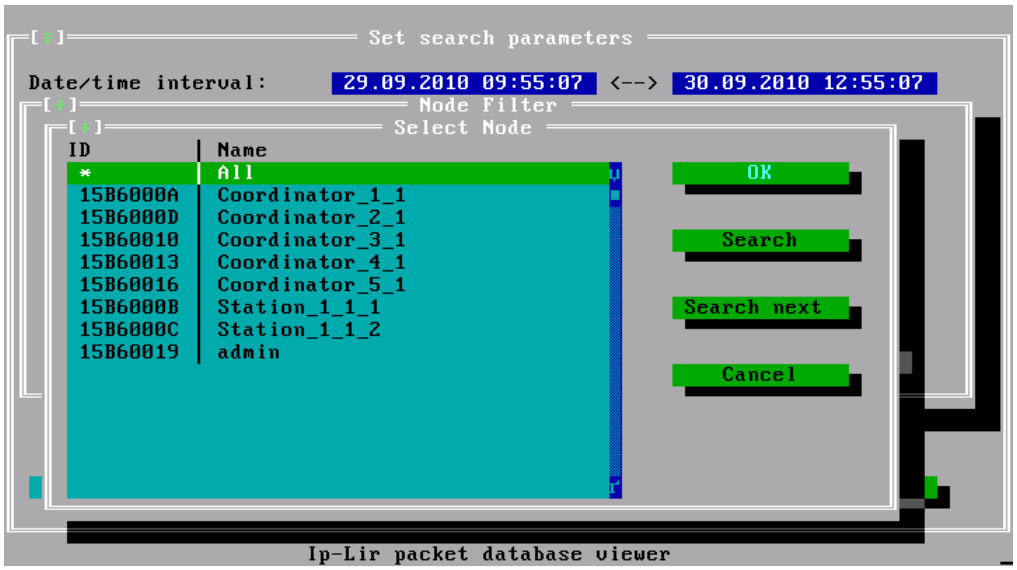

Рисунок 17. Список защищенных узлов для выбора отправителя или получателя

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 15 на стр. 134). Для завершения просмотра журнала нажмите кнопку **Exit**.

Список записей, найденных в журнале регистрации IP-пакетов, выводится в окне **View results** (см. Рисунок 18 на стр. 138). 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.
- Локальный 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.
- Шестнадцатеричный идентификатор протокола Ethernet.
- 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.

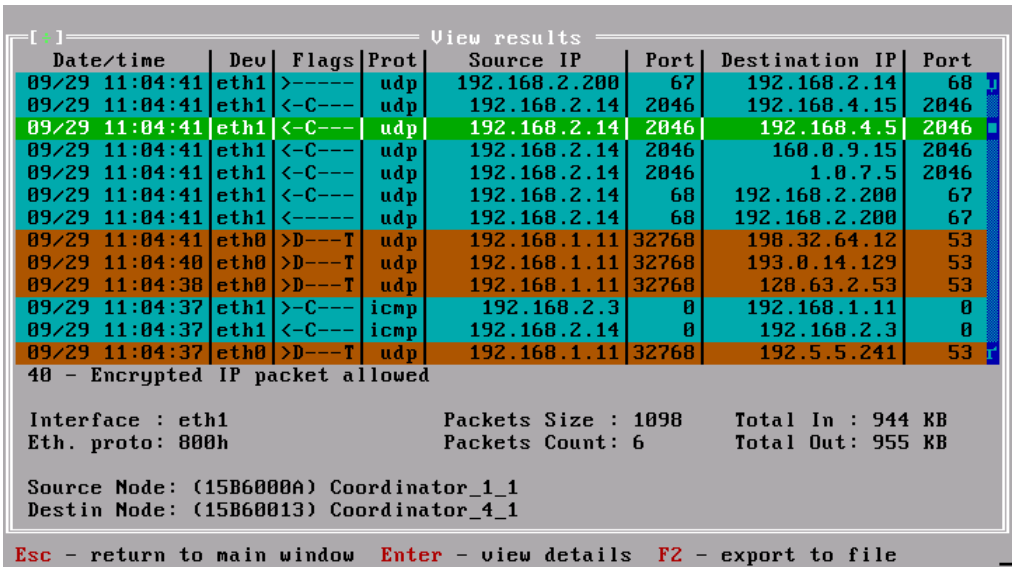

Рисунок 18. Вывод списка найденных записей

Для любой выбранной в списке записи можно просмотреть более детальную информацию, для этого нажмите клавишу **Enter**.

```
[+]                                Record details
Events: 30 - Local IP packet blocked by Public Network filter

Interval Begin:  04.03.2015 16:49:31
           End:  04.03.2015 16:49:33

Interface: eth0      Ethernet protocol: 800h
Size:      234 B     Count:             3

Drop:      YES       Encrypted NO
Direction: Incoming  NAT:      NO
Broadcast: NO        Forward:  NO

IP protocol:    17 - UDP (User Datagram)
Source IP:      10.0.17.19        Port: 137
Destination IP: 10.0.17.255       Port: 137

Key number:                 00000000
Source Node                 00000000

Destination Node            00000000

Esc  or  Enter - return to view results
```

Рисунок 19. Детальная информация о событии

Полученный список пакетов можно экспортировать в текстовый файл, для этого в окне **View Results** нажмите кнопку **F2**. При этом предлагается выбрать файл, в который будет экспортирован журнал, или ввести его имя вручную. В файл экспорта записываются все поля, которые выводятся в окне **View Results**.

Информация о размере пакетов поддерживается, начиная с версии 2.8-157. При установке этой или более поздней версии поверх более старой журналы пакетов преобразуются к новому формату, однако информация о размере для пакетов, записанных старой версией, будет недоступна и отображается как **N/A** (Not available). При удаленном просмотре журнала более старой версии информация о размере пакетов также недоступна и не показывается. Это происходит и при удаленном просмотре журнала новой версии с компьютера, где установлена старая версия.

В нижней части окна со списком найденных в журнале записей (см. Рисунок 18 на стр. 138) 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 Кбайт, то размер отображается в байтах и к числу добавляется суффикс «B»;
- если суммарный размер больше 100 Кбайт, но меньше 100 Мбайт, то размер отображается в килобайтах и к числу добавляется суффикс «KB»;
- если суммарный размер больше 100 Мбайт, то размер отображается в мегабайтах и к числу добавляется суффикс «MB».

Программа просмотра информации о защищенном узле

Программа `iplir_remote_info` предназначена для просмотра системной и статистической информации, получаемой от локального или удаленного защищенного узла. Системная информация включает в себя версию управляющего демона и драйвера, установленных на компьютере, идентификатор и имя узла, общие сведения о каждом интерфейсе (IP-адрес и другие параметры). Статистическая информация включает в себя информацию об обработке входящих и исходящих IP-пакетов для заданного интерфейса.

Примечание. Слово `Disabled`, присутствующее в информации об интерфейсе, указывает на его недоступность. Для недоступных интерфейсов не выводится информация об IP-адресах и статистическая информация.

Программа называется `iplir_remote_info` и помещается после установки в каталог `/sbin`. Она запускается из управляющего скрипта `iplir` командами `iplir info` и `iplir ifinfo`. При запуске программа считывает конфигурационный файл `/etc/iplirnetpsw`. В первой строке этого файла должно быть указано доменное имя или IP-адрес узла, информацию о котором нужно просматривать. Остальные строки игнорируются. По умолчанию после инсталляции ПО файл `/etc/iplirnetpsw` содержит одну строку с адресом 127.0.0.1, то есть предполагается просмотр информации о себе. Если в файле `/etc/iplirnetpsw` указан IP-адрес удаленного узла, то после запуска программа запрашивает пароль администратора сети ViPNet.

По команде `iplir info` программа выводит системную информацию и статистическую информацию для первого сетевого интерфейса в системе. Выводится следующая статистическая информация:

- `Non-encrypted packets passed` (Обработано незашифрованных IP-пакетов) – количество принятых и отправленных незашифрованных пакетов.
- `Non-encrypted packets dropped` (Блокировано незашифрованных IP-пакетов) – количество входящих и исходящих блокированных незашифрованных пакетов.
- `Encrypted packets passed` (Обработано зашифрованных IP-пакетов) – количество принятых расшифрованных и отправленных зашифрованных пакетов.
- `Encrypted packets dropped` (Блокировано зашифрованных IP-пакетов) – количество входящих и исходящих блокированных зашифрованных пакетов.
- `Non-encrypted broadcast packets passed` (Обработано незашифрованных широковещательных сообщений) – количество принятых и отправленных незашифрованных широковещательных сообщений.
- `Non-encrypted broadcast packets dropped` (Блокировано незашифрованных широковещательных сообщений) – количество входящих и исходящих блокированных незашифрованных широковещательных сообщений.

- `Encrypted broadcast packets passed` (Обработано зашифрованных широковещательных сообщений) – количество принятых и отправленных зашифрованных широковещательных сообщений.
- `Encrypted broadcast packets dropped` (Блокировано зашифрованных широковещательных сообщений) – количество входящих и исходящих блокированных зашифрованных широковещательных сообщений.

Для просмотра информации о других сетевых интерфейсах (в том случае, если на компьютере установлено несколько сетевых адаптеров) необходимо использовать команду `iplir ifinfo`. При этом вначале надо использовать команду `iplir info`, которая показывает список интерфейсов узла, и затем использовать команду `iplir ifinfo`, задав в ней имя нужного интерфейса, например `iplir ifinfo eth1`.

Адрес узла, информацию о котором нужно получить, можно задать не только в файле `/etc/iplirnetpsw`, но также непосредственно в командах `iplir info` и `iplir ifinfo`. В команде `iplir ifinfo` адрес узла должен быть указан после имени интерфейса, например `iplir ifinfo eth1 10.0.2.40`.

Программа просмотра информации о состоянии системы защиты от сбоев

Программа `failover_info` предназначена для просмотра информации о текущем состоянии системы защиты от сбоев (см. «Система защиты от сбоев» на стр. 126). Данная информация включает в себя:

- версию ПО ViPNet Coordinator Linux и демона failoverd, установленных на узле;
- идентификатор и имя узла;
- режим работы системы защиты от сбоев (одиночный или режим кластера горячего резервирования серверов);
- локальное время на узле;
- общую загрузку CPU;
- текущую информацию о состоянии управляющего демона, демонов mftpd, algd, axis2.cgi и failoverd.

Программа называется `failover_info` и помещается после инсталляции ПО в каталог `/sbin`. Обычно она запускается из управляющего скрипта `failover` командой `failover info`. При таком запуске программа считывает конфигурационный файл `/etc/iplirnetpsw`. По команде `failover info` программа выводит информацию на текущую консоль.

Если демон failoverd остановлен, то выводится только сообщение о режиме работы:

```
Failover is in <single|cluster> mode
```

Если демон failoverd запущен, то при работе в одиночном режиме выводится следующая информация:

```
Versions: ViPNet 3.6.0 (475), daemon 1.3 (14)
Workstation configured for ID 10E1079E (Vipnux-_1_3)
The workstation works in a single mode of protection against failures
Workstation time (utc: 1174558030) Thu Mar 25 13:07:10 2010
```

`failover mode` * `single` — режим работы системы защиты от сбоев;

`failover uptime` * `6d  0:23` — время работы демона failoverd;

`total cpu` * `0%` — общая загрузка CPU в системе;

`failover state` * `works` — состояние демона failoverd;

`failover cpu` * `0%` — загрузка CPU демоном failoverd;

`iplir state` * `works` — состояние управляющего демона iplircfg;

`iplir cpu` * `0%` — загрузка CPU демоном iplircfg;

`mftp state` * `works` — состояние демона mftpd;

`mftp cpu` * `0%` — загрузка CPU демоном mftpd;

`alg state` * `works` — состояние демона algd;

`alg cpu` * `0%` — загрузка CPU демоном algd;

`webgui state` * `works` — состояние демона axis2.cgi;

`webgui cpu` * `0%` — загрузка CPU демоном axis2.cgi.

При выводе информации используются следующие обозначения режимов:

- `single` – одиночный режим работы;
- `active` – активный режим кластера горячего резервирования серверов;
- `passive` – пассивный режим кластера горячего резервирования серверов.

Используемые обозначения состояний:

- `works` – приложение работает корректно (с точки зрения системы защиты от сбоев);
- `stopped` – приложение остановлено пользователем;
- `unknown` – состояние приложения неизвестно. Данное состояние может быть определено в случае, если был идентифицирован сбой контролируемого приложения, попытка его перезапуска, но данных о его корректном старте пока нет.

Формат вывода информации при работе в режиме кластера горячего резервирования серверов приведен в документе «ViPNet Coordinator Linux. Система защиты от сбоев. Руководство администратора».

Программа смены пароля пользователя

Программа `iplirpasswd` предназначена для смены пароля пользователя ViPNet. Поскольку пароли при начальной установке назначаются администратором сети ViPNet, у пользователя может возникнуть необходимость изменить пароль после установки.

Программа называется `iplirpasswd` и помещается после установки в каталог `/sbin`. Обычно она запускается из управляющего скрипта `iplir` командой `iplir passwd`. Перед сменой пароля необходимо остановить службы ViPNet (управляющий демон iplircfg, демон системы защиты от сбоев failoverd, транспортный модуль mftpd, демон обработки прикладных протоколов algd). После смены пароля необходимо запустить службы ViPNet.

При смене пароля программа запрашивает у пользователя старый пароль. Если введенный пароль неверен, программа завершает свою работу. Если пароль верен, пользователю предлагается ввести новый пароль в режиме экранирования символов, т.е. при вводе символы пароля не отображаются на экране. В случае успешной смены пароля программа выводит соответствующее сообщение и завершает свою работу. При этом новый пароль шифруется (независимо от того, был или не был зашифрован старый пароль) и сохраняется в файле `/etc/iplirpsw` в зашифрованном виде.

Программа распаковки дистрибутива ключей

Программа `unmerge` предназначена для распаковки дистрибутива ключей — файла с расширением `.dst`, переданного администратору сетевого узла администратором сети ViPNet. После инсталляции ПО эта программа помещается в каталог `/sbin`.

Программа `unmerge` имеет следующий формат:

`unmerge [-f] <файл_дистрибутива> <каталог_распаковки>`, где

`-f` – необязательный ключ для распаковки без подтверждения очистки каталога;

`<файл_дистрибутива>` – файл, содержащий дистрибутив ключей;

`<каталог_распаковки>` – каталог для распаковки.

Программа `unmerge` распаковывает заданный файл дистрибутива в заданный каталог. Если каталог не существует, то он автоматически создается.

Если каталог существует и он не пустой, то выдается соответствующее предупреждение и запрашивается подтверждение на продолжение. В случае согласия перед распаковкой из указанного каталога удаляются текущие справочники и ключи, прочие данные остаются. В случае отказа программа завершает работу.

Если указан необязательный ключ `-f`, то подтверждение на очистку каталога не запрашивается — программа автоматически удаляет из каталога текущие справочники и ключи.

Программа работы с конфигурациями ViPNet

В настоящее время конфигурационные файлы, адресные справочники и ключи в ViPNet Coordinator Linux никак не защищены от случайного искажения в них информации в результате ошибок ПО или некорректных действий пользователя. В частности, это может привести к тому, что в некоторых случаях после обновления справочников и (или) ключей конфигурационные файлы искажаются, и работа системы нарушается. Во избежание таких ситуаций в состав ViPNet Coordinator Linux включена система сохранения файлов конфигурации, справочников и ключей (в дальнейшем — конфигурация) таким образом, чтобы пользователь мог сохранить текущую конфигурацию, загрузить одну из ранее сохраненных или удалить имеющуюся сохраненную конфигурацию. При этом перед проведением обновления в зависимости от настроек транспортного модуля MFTP текущая конфигурация может быть сохранена автоматически (автосохраненная конфигурация). Загрузка сохраненной пользователем конфигурации происходит только вручную по команде пользователя, при этом текущая конфигурация теряется. Однако перед загрузкой пользователю предлагается сохранить текущую конфигурацию для возможного использования в будущем. Удаление сохраненных пользователем конфигураций также производится только вручную.

Существует два типа конфигурации:

- **Частичная конфигурация (partial)**.

  Частичная конфигурация включает в себя все конфигурационные файлы компонентов ViPNet, в том числе файлы `iplir.conf`, `mftp.conf`, `failover.ini`, `policy.conf`.

- **Полная конфигурация (full)**.

  Полная конфигурация включает в себя все файлы частичной конфигурации, а также все справочники и ключи. В полную конфигурацию не включаются: временные служебные файлы, файлы журналов и очередь конвертов MFTP.

Для управления конфигурациями в состав ПО ViPNet Coordinator Linux входит специальная программа `/sbin/iplir-config-manager`. Она запускается из управляющего скрипта `iplir`, расположенного в каталоге `/sbin`.

Внимание! Непосредственный вызов программы `iplir-config-manager` пользователем крайне нежелателен!

Для работы с конфигурациями предназначены следующие команды:

- `iplir 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 21.03.2008 at 11:49, never loaded
"Config_2", part, saved on 21.03.2008 at 11:49, never loaded
"autosave-2005-03-24-17-05-31", full, saved on 24.03.2008 at 17:05, loaded at 01.04.2005 at 12:45
"rollback-2006-10-12", version 2.8.16, part, saved on 12.10.2006 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`. В случае rollback-конфигурации имя конфигурации начинается со слова `rollback`.

- `iplir config save <имя> [<тип>]` – сохранение текущей конфигурации заданного типа.

  `имя` – имя, под которым нужно сохранить конфигурацию.

  `тип` – тип конфигурации (`full` или `partial`, по умолчанию `full`).

  Если уже существует сохраненная конфигурация с заданным именем, созданная текущей установленной версией ПО, то программа запрашивает пользователя, нужно ли перезаписать имеющуюся конфигурацию.

Внимание! В именах конфигураций можно использовать только символы латинского алфавита, цифры, знаки дефис и подчеркивание.

- `iplir config load <имя> [<версия>]` – восстановление конфигурации с заданным именем `имя` и версией `версия`. Перед восстановлением конфигурации программа 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, то команда относится только к сохраненным конфигурациям без информации о версии ПО. Если имеется несколько конфигураций с одинаковыми именами, но различными версиями ПО, то для однозначной идентификации восстанавливаемой конфигурации данный параметр необходимо указать, иначе программа выдаст соответствующее сообщение об ошибке.

При попытке восстановления конфигурации программа сравнивает текущую версию ПО ViPNet и версию, к которой относится конфигурация. Если текущая версия ПО равна или больше версии, к которой относится конфигурация, то производится восстановление без дополнительного подтверждения. В противном случае пользователю сообщается, что конфигурация относится к более новой версии ПО, и, возможно, потребуется 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.

- `iplir config delete <имя> [<версия>]` – удаление сохраненной конфигурации с заданным именем `имя`. В качестве имени можно указывать символы подстановки (wildcards) «*» и «?», которые обозначают любое количество символов и любой один символ соответственно. Таким образом, можно производить удаление конфигураций по маске имени. Например:

  ```
  iplir config delete "Config_*"
  iplir config delete "autosave-2005-08-??-*" 2.8.16
  ```

  Если под заданный шаблон имени подходит более одной конфигурации, то выдается предупреждение и у пользователя запрашивается подтверждение на удаление.

Внимание! При использовании символов подстановки они должны обязательно экранироваться с помощью двойных кавычек. В противном случае команда может быть неправильно интерпретирована оболочкой (shell) операционной системы, что может привести 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, то команда относится только к сохраненным конфигурациям без информации о версии ПО. В случае, если параметр `версия` не указан, а параметр `имя` указан в виде маски (используются символы подстановки), будут удалены все конфигурации, попадающие под заданную маску имени, независимо от версии ПО.

Перед проведением операций сохранения и восстановления конфигураций должны быть остановлены все службы ViPNet. Если это условие не выполняется, то скрипт `iplir` выдаст соответствующее сообщение и не будет выполнять никаких действий.

10

Тонкая настройка ПО ViPNet Coordinator Linux

Для чего нужна тонкая настройка 150

Изменение основных номеров устройств, используемых драйверами 151

Выбор режима шифрования 152

Фрагментирование шифрованных ViPNet-пакетов 153

Настройка максимального размера очереди пакетов в драйвере 154

Управление числом потоков в драйвере 155

Выбор метода подсчета контрольной суммы пакета 157

Настройка системного времени ОС Linux при использовании ПО ViPNet 158

Ручное переназначение виртуальных адресов узлов 159

Для чего нужна тонкая настройка

Сведения о тонкой настройке предназначены только для квалифицированных системных администраторов, которым может потребоваться настроить ViPNet на работу в различных нестандартных условиях.

В состав ViPNet входят следующие драйверы:

- `drviplir` – основной сетевой драйвер;
- `itcswd` – драйвер watchdog;
- `itcscrpt` – криптографический драйвер.

В некоторых ситуациях, описанных ниже, требуется, чтобы какой-либо из этих драйверов работал в режиме, отличном от режима по умолчанию. Это делается путем передачи драйверу параметров при его загрузке командой `insmod`. Имя параметра для передачи драйверу имеет вид: `<имя_драйвера>_<имя_параметра>`. После имени параметра следует знак равенства, после которого указывается значение параметра.

Для загрузки драйверов ViPNet с параметрами следует добавить нужный параметр в переменную, содержащую набор параметров соответствующего драйвера. Эти переменные задаются в файле `/etc/iplirSKnums`. Для драйвера `drviplir` параметры задаются в переменной `DRVIPLIR_PARAMS`, для драйвера `itcswd` — в переменной `ITCSWD_PARAMS`, для драйвера `itcscrpt` — в переменной `ITCSCRPT_PARAMS`.

Изменение основных номеров устройств, используемых драйверами

Все драйверы ViPNet используют динамически выделяемые основные номера устройств (`major device number`). При этом, получив динамический номер, они самостоятельно создают специальный файл (`device node`), посредством которого прикладные программы обращаются к драйверу. Таким образом, прикладные программы никак не зависят от используемого драйвером основного номера устройства.

Однако при совместной работе драйверов ViPNet с другими драйверами, которые не используют механизм динамического выделения основных номеров устройств, может получиться так, что динамически выделенный драйверу ViPNet номер совпадет с номером, используемым другим драйвером. Если обнаруживается такая ситуация, то существует возможность задать драйверам ViPNet статические номера устройств, которые они будут использовать. Для задания статических номеров устройств используется параметр `<имя_драйвера>_major`. Значением такого параметра должен быть основной номер устройства, который должен использовать драйвер. Например, если требуется загрузить драйвер `drviplir` так, чтобы он использовал номер 250, необходимо добавить в переменную `DRVIPLIR_PARAMS` следующую запись: `drviplir_major=250`. Разумеется, нужно задавать всем драйверам ViPNet разные номера.

Выбор режима шифрования

В предыдущих версиях ПО ViPNet для криптографических операций использовался только один режим применения алгоритма ГОСТ 28147-89 — режим шифрования CFB (Cipher Feedback). Начиная с версии 3.7.1, реализован еще один режим шифрования — CTR (Counter mode). Режим CTR позволяет получить выигрыш в скорости до 20% по сравнению с режимом CFB. По умолчанию в криптографическом драйвере по-прежнему используется режим CFB, однако есть возможность выбрать режим CTR.

Для выбора режима шифрования используется параметр `use_gost_ctr`, задаваемый в переменной `ITCSCRPT_PARAMS`. Параметр может принимать следующие значения:

- `0` — режим CFB (значение по умолчанию);
- `1` — режим CTR.

При выборе режима шифрования следует учитывать следующие ограничения:

- Режим CTR работает только на процессорах, поддерживающих SSSE3 (Supplemental Streaming SIMD Extensions 3). Если процессор не поддерживает SSSE3, то автоматически будет выбран режим CFB независимо от значения, заданного в параметре `use_gost_ctr`.
- При непосредственном взаимодействии двух сетевых узлов с разными режимами шифрования на каждом узле автоматически будет использоваться режим CFB.
- На координаторе, через который взаимодействуют сетевые узлы с режимом CTR, должно быть установлено ПО ViPNet Coordinator Linux версии 3.7.1 и выше, которое поддерживает этот режим, чтобы обеспечить прохождение транзитного трафика.

Фрагментирование шифрованных ViPNet-пакетов

В предыдущих версиях ПО драйвер ViPNet уменьшал размер MTU (Maximum Transmission Unit — максимальный размер блока данных в байтах, который может быть передан на канальном уровне) на всех интерфейсах, которые обрабатываются ViPNet. Это делалось во избежание фрагментирования пакетов вследствие добавления служебных данных ViPNet.

Начиная с версии 2.8-284, драйвер ViPNet использует новую технологию управления MSS (Maximum Segment Size — максимальный размер блока данных в байтах, который может быть передан по протоколу TCP) для TCP-соединений, которая позволяет избежать фрагментирования пакетов без изменения MTU на интерфейсе. Поэтому параметр `drviplir_mtudiff` больше не используется драйвером. При обновлении старой версии ПО ViPNet данный параметр удаляется из файла `/etc/iplirSKnums`.

Настройка максимального размера очереди пакетов в драйвере

В процессе обработки сетевых пакетов драйвер ViPNet использует специальную очередь, в которую помещаются все входящие и исходящие пакеты со всех сетевых интерфейсов. По мере обработки пакетов размер очереди может как увеличиваться, так и уменьшаться в зависимости от интенсивности трафика, количества интерфейсов, производительности компьютера и так далее. Чтобы избежать лишнего расхода оперативной памяти, приводящего к нестабильной работе системы, размер очереди имеет ограничение, которое по умолчанию равно 2048. Если в очереди содержится максимальное количество пакетов, то остальные входящие и исходящие пакеты будут блокироваться до тех пор, пока в очереди не появится свободное место. При достижении размера очереди 90% от максимального драйвер информирует пользователя об этом специальным сообщением. В типичных случаях сетевого трафика вероятность возникновения таких ситуаций очень мала. Однако в некоторых частных случаях, например при достаточно интенсивном UDP-трафике, очередь может переполняться. Для предотвращения появления таких ситуаций предусмотрен параметр `drviplir_pqsize`, позволяющий изменять ограничение на размер очереди драйвера. Данный параметр нужно добавить в переменную `DRVIPLIR_PARAMS` файла `/etc/iplirSKnums`, например: `drviplir_pqsize=4096`. В этом случае ограничение на размер очереди станет равным 4096.

Управление числом потоков в драйвере

Начиная с версии 3.6.0, в драйвере ViPNet поддерживается режим многопоточной обработки трафика (шифрование и фильтрация) с использованием сразу нескольких процессоров. Это позволяет задействовать возможности многопроцессорных систем и существенно увеличить скорость обработки.

Включение и отключение многопоточного режима осуществляется путем установки числа потоков. Максимальное число потоков, которое может быть установлено, равно количеству процессоров, умноженному на 4, но не более 32.

Для управления потоками обработки сетевого трафика используется параметр `threads_count`, задаваемый в переменной `DRVIPLIR_PARAMS`. В этом параметре можно явно указать число потоков. При отсутствии в переменной `DRVIPLIR_PARAMS` параметра `threads_count` число потоков будет равно количеству процессоров в системе. Если количество потоков, заданных в данном параметре, меньше либо равно нулю или больше максимального, то во время загрузки драйвера iplir количество потоков автоматически станет равно количеству процессоров.

Примечание. В более ранних версиях ПО по умолчанию устанавливался однопоточный режим (`threads_count=1`).

При переходе на текущую версию ПО с версий до 3.7.0:

- если ранее в параметре `threads_count` было задано число потоков, отличное от 1, то значение параметра не изменяется;
- если ранее был установлен однопоточный режим, то автоматически устанавливается многопоточный режим (параметр `threads_count` удаляется из переменной `DRVIPLIR_PARAMS`).

Также в работающем драйвере число потоков можно менять с помощью следующей команды:

```
echo X > /proc/driver/drviplir/thread_count
```

где `X` — желаемое число потоков от 1 до максимального числа потоков.

Дополнительно можно задать маску процессоров, которые будут участвовать в обработке трафика. Маска задается в переменной `DRVIPLIR_PARAMS`, например, с помощью следующей команды:

```
DRVIPLIR_PARAMS='cpu_use_mask="XX..X"'
```

где `XX..X` — двоичная последовательность, каждый разряд которой соответствует определенному процессору. Количество разрядов равно количеству процессоров в системе. Нумерация процессоров при задании маски начинается с нуля (cpu0, cpu1, cpu2 и так далее). Единицы в маске означают, что процессор участвует в обработке трафика, нули — что не участвует.

Например, в системе имеется четыре процессора cpu0, cpu1, cpu2, cpu3, и вам необходимо, чтобы в обработке трафика участвовали только процессоры cpu0, cpu1, cpu3, а процессор cpu2 не должен обрабатывать трафик. Для этого необходимо задать следующую маску:

```
DRVIPLIR_PARAMS='cpu_use_mask="1101"'
```

В результате процессор cpu2 не будет участвовать в процессе обработки трафика.

Если задано число потоков больше, чем имеется процессоров в системе, и указана маска, то потоки будут накладываться на маску циклически.

При включении многопоточного режима следует иметь в виду, что в случае интенсивного сетевого трафика все ресурсы могут быть задействованы в обработке трафика, и на другие пользовательские приложения их может не хватить. В этом случае рекомендуется уменьшить число потоков, чтобы освободить часть ресурсов для выполнения других задач.

Выбор метода подсчета контрольной суммы пакета

Вычисление контрольной суммы необходимо для контроля целостности пакетов, а также для подтверждения того, что пакеты не были изменены при передаче.

В ПО ViPNet Coordinator Linux версии 3.7.2 и выше можно использовать два метода подсчета контрольной суммы IP-пакета:

- Частично аппаратный метод. В данном методе контрольная сумма служебного заголовка вычисляется программными средствами, а вычисление контрольной суммы остального содержимого пакета осуществляется средствами ядра Linux либо сетевой карты.
- Полностью программный метод. В данном методе вычисление контрольной суммы всего IP-пакета (в том числе служебного заголовка) осуществляется программными средствами.

  Использовать полностью программный метод подсчета контрольной суммы рекомендуется, если возникли какие-либо проблемы при подсчете частично аппаратным методом.

Для выбора метода подсчета контрольной суммы пакета используется параметр `hw_csum`, задаваемый в переменной `DRVIPLIR_PARAMS`. Параметр может принимать следующие значения:

- `1` — первый метод подсчета контрольной суммы (частично аппаратный). Данное значение используется по умолчанию;
- `0` — второй метод подсчета (полностью программный).

Настройка системного времени ОС Linux при использовании ПО ViPNet

Если в процессе работы ПО ViPNet вам потребовалось изменить системное время, выполните следующие действия:

1. Перед изменением остановите следующие службы ViPNet:
   - демон системы защиты от сбоев (`failoverd`);
   - управляющий демон (`iplircfg`);
   - транспортный модуль (`mftpd`);
   - демон обработки прикладных протоколов (`algd`).
2. Установите новое системное время.
3. После установки нового времени снова запустите службы ViPNet, перечисленные выше.

Данные службы периодически считывают и используют при анализе текущее системное время, поэтому если изменить время в процессе их функционирования, то может нарушиться логика работы отдельных подсистем. Наиболее опасен в данной ситуации перевод системных часов назад.

Ручное переназначение виртуальных адресов узлов

В стандартном режиме работы управляющий демон автоматически назначает виртуальные адреса (см. «Принципы назначения виртуальных адресов» на стр. 64) 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. 61), 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.

11

Протоколирование событий, ведение и просмотр журналов

Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP 161

Контроль дампов аварийного завершения программы ViPNet Linux 164

Журналы устранения неполадок ПО ViPNet Coordinator Linux 166

Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP

ViPNet Coordinator Linux имеет возможность взаимодействия с сервером SNMP, что позволяет удаленно получать информацию о времени работы системы, интерфейсах, количестве пропущенных и блокированных пакетов и так далее, а также уведомлять удаленную станцию сетевого менеджмента (NMS) о наиболее важных событиях в системе.

В настоящее время ПО ViPNet взаимодействует только с сервером `ucd-snmp`, который распространяется свободно и поставляется вместе с ViPNet Coordinator Linux. Исходный код этого сервера, а также все необходимые для работы подсистемы SNMP файлы находятся в дистрибутиве ПО ViPNet Coordinator Linux в подкаталоге `snmp`.

Чтобы включить поддержку SNMP, выполните следующее:

- Если на компьютере уже установлен какой-либо демон SNMP, удалите его.
- Войдите в каталог `distribute`, где находится распакованный дистрибутив ключей, и распакуйте файл архива ПО командой `tar xvfz distribute.tar.gz`.
- Запустите скрипт `install-snmp` из подкаталога `snmp` каталога дистрибутива. Это необходимо делать с правами пользователя `root`. Этот сценарий компилирует демон SNMP таким образом, чтобы в нем была доступна статистика ViPNet, и устанавливает его в `/usr/local/sbin`.

  Для успешной компиляции необходимо, чтобы на компьютере были установлены все программные пакеты, которые перечислены в системных требованиях (см. «Системные требования» на стр. 12). В процессе компиляции пользователю задаются вопросы об электронном адресе (e-mail) администратора, местонахождении системы и тому подобные – на них нужно ответить соответствующим образом.
- Если планируется использовать оповещения (SNMP Traps), то после установки создайте (если он отсутствует) и отредактируйте файл `/usr/local/share/snmp/snmpd.conf`, добавив в него следующие строки:

  ```
  informsink <хост-получатель>
  trapsink <хост-получатель>
  trap2sink <хост-получатель>
  ```

  где вместо `<хост-получатель>` укажите имя или IP-адрес компьютера, который будет получать оповещения.
- Запустите `/usr/local/sbin/snmpd`. Если требуется запускать демон snmpd каждый раз при загрузке системы, то необходимо настроить запуск вручную – установочный сценарий не делает этого.

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 ПО 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):

- .1.3.6.1.4.1.10812.1.1.1 — 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 ПО ViPNet, при перезапуске управляющего демона значение обнуляется;
- .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 блокированных 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 блокированных исходящих нешифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.8 — число пропущенных входящих шифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.9 — число блокированных входящих шифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.10 — число пропущенных исходящих шифрованных пакетов;
  - .1.3.6.1.4.1.10812.1.2.2.1.11 — число блокированных исходящих шифрованных пакетов.

ПО ViPNet использует следующие оповещения (SNMP Traps):

- .1.3.6.1.4.1.10812.1.0.1 — запуск управляющего демона;
- .1.3.6.1.4.1.10812.1.0.2 — остановка управляющего демона;
- .1.3.6.1.4.1.10812.1.0.3 — запуск демона mftpd;
- .1.3.6.1.4.1.10812.1.0.4 — остановка демона mftpd;

- .1.3.6.1.4.1.10812.1.0.7 — запуск демона failoverd (только в активном режиме);
- .1.3.6.1.4.1.10812.1.0.8 — остановка демона failoverd (только в активном режиме);
- .1.3.6.1.4.1.10812.1.0.9 — переключение демона failoverd из пассивного режима в активный.

Как правило, NMS позволяет настроить определенную реакцию на каждое из оповещений (отсылка email, sms и так далее).

Контроль дампов аварийного завершения программы ViPNet Linux

Все ОС Linux предоставляют возможность создания дампов программ при их аварийном завершении (`core files`). Как правило, аварийное завершение программ происходит довольно редко, поэтому в случае возникновения такой ситуации необходимо отправить сохраненные операционной системой дампы в адрес разработчика. Эти дампы содержат важную информацию, необходимую разработчикам для обнаружения и устранения причин аварийного завершения программ, увеличения надежности и стабильности функционирования ПО ViPNet.

Начиная с версии 2.8-284 программы, входящие в состав ПО ViPNet Coordinator Linux, также автоматически создают дампы аварийного завершения и контролируют их наличие.

Программы, для которых создаются и проверяются `core`-файлы:

- `iplircfg`;
- `dbviewer`;
- `iplir_remote_info`;
- `iplir-config-manager`;
- `failoverd`;
- `iplirpasswd`;
- `mftpd`;
- `mailtrans`;
- `mftp_remote_info`;
- `algd`;
- `axis2.cgi`.

В каталоге, в котором хранятся справочники и ключи, создается каталог `coredumps`, внутри которого каждая из указанных программ создает подкаталог, соответствующий ее имени. При старте программы проверяется наличие соответствующего каталога, и если он отсутствует, то такой каталог автоматически создается. После этого проверяется наличие `core`-файлов для всех программ. Если для какой-либо из программ ПО ViPNet Coordinator Linux были найдены `core`-файлы, то выдается соответствующее сообщение на терминал и в syslog с просьбой выслать их в адрес разработчика с указанием версии ПО. После отсылки `core`-файлы можно удалить, чтобы освободить место на диске.

Для создания дампов аварийного завершения требуется некоторое свободное место в разделе, где расположен каталог со справочниками и ключами. В случае недостатка свободного места `core`-

файлы создаваться не будут. Во избежание таких ситуаций при старте вышеперечисленных программ производится проверка свободного места на диске. Если места в соответствующем разделе меньше, чем требуется, то выдается сообщение в syslog (и на терминал, если это старт программы). Кроме того, проверка свободного места производится каждые 5 минут в процессе работы управляющего демона (`iplircfg`).

Журналы устранения неполадок ПО ViPNet Coordinator Linux

В процессе работы службы (демоны), входящие в состав ViPNet Coordinator Linux, 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.

Журналы устранения неполадок могут записываться и храниться различными способами:

- В виде текстового файла в указанном настройками каталоге (этот способ используется по умолчанию).
- В общем системном журнале ОС Linux, запись в который осуществляется специальной службой – демоном syslogd.

Настройка протоколирования осуществляется с помощью параметров, задаваемых в секции `[debug]` конфигурационных файлов демонов. В секции `[debug]` задаются следующие параметры:

- `debuglogfile` – место хранения журнала.

  Этот параметр имеет следующий формат: `<спецификатор протокола>:<спецификатор URL для данного протокола>`. Текущая версия, как было сказано выше, поддерживает два протокола ведения журналов:

  - Запись в файл. В этом случае спецификатор протокола должен иметь значение `file`, а спецификатор URL должен содержать полный путь к файлу журнала, например:

    ```
    debuglogfile= file:/var/log/iplircfg.debug
    ```

    При данных настройках журнал устранения неполадок для управляющего демона будет записываться в указанный файл, как и в предыдущих версиях ПО ViPNet Coordinator Linux.

  - Запись в системный журнал ОС Linux. В этом случае спецификатор протокола должен иметь значение `syslog`, а спецификатор URL должен состоять из двух частей, разделенных точкой и определяющих источник (`facility`) и важность (`level`) для сообщений в системном журнале, например:

    ```
    debuglogfile= syslog:daemon.debug
    ```

    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 критичности сообщений для администратора ОС Linux и может принимать, в частности, следующие значения:

- `err` (ошибка);
- `info` (информационное сообщение);
- `debug` (отладочное сообщение).

При использовании спецификатора протокола `syslog` с заданными `facility` и `level` сообщения журнала устранения неполадок будут передаваться системному демону syslogd, который будет обрабатывать их в соответствии со своими настройками и сохранять в системном журнале. Помимо сохранения сообщений в локальных файлах системного журнала существует возможность настройки демона syslogd для передачи сообщений на удаленный сервер. Подробнее с возможностями и способами настройки демона syslogd можно ознакомиться в соответствующем справочном руководстве по syslogd, поставляемом в составе используемого дистрибутива Linux.

- `debuglevel` – уровень протоколирования.

  Уровень протоколирования определяет степень детализации информации, выводимой соответствующей службой в журнал, и задается числом от -1 до 5. С повышением уровня протоколирования увеличивается количество информации, помещаемой в файл журнала. Как и место хранения, уровень протоколирования каждого демона задается в его собственном конфигурационном файле (см. «Настройка параметров защищенной сети» на стр. 49) в зависимости от требуемой степени детализации получаемой информации. По умолчанию протоколирование ведется на 3-ем уровне, который соответствует среднему уровню детализации. В большинстве случаев данной информации достаточно для диагностики работы службы. Однако в некоторых случаях, при возникновении проблем в работе ПО, требуется более подробная информация, что требует установки более высокого уровня протоколирования. В этих случаях рекомендуется обратиться в службу поддержки компании «ИнфоТеКС» за консультацией по выбору оптимального уровня протоколирования, необходимого для исследования и устранения неполадок.

Как было сказано, увеличение уровня протоколирования приводит к увеличению объема информации в файле, в котором данный журнал хранится, и соответственно размера данного файла. При использовании уровня протоколирования по умолчанию (3) размер файла журнала увеличивается в среднем на 150-300 Кбайт в час в зависимости от интенсивности трафика, количества узлов в сети и так далее. При более высоких уровнях протоколирования размер файла журнала растет быстрее. В случае хранения журналов служб ViPNet в системном журнале проблем с превышением максимального размера файла журнала в большинстве случаев быть не должно, так как в ОС Linux по умолчанию настроена ротация файлов системного журнала. Однако в случае хранения журналов служб ViPNet в отдельных файлах (с использованием спецификатора протокола `file` в параметре `debuglogfile`) такая проблема может возникнуть.

Для предотвращения превышения максимального размера файла хранения журнала и критического уменьшения свободного места файловой системы ПО ViPNet Coordinator Linux реализует механизм ротации и архивирования журналов. Для этого используется системная утилита `logrotate`, являющаяся базовым средством ротации журналов в ОС Linux. Эта утилита

обычно запускается на регулярной основе с помощью планировщика `cron`. Рекомендуется запускать утилиту раз в день, сконфигурировав `cron` соответствующим образом.

Внимание! Наличие утилиты `logrotate`, а также планировщика `cron` является обязательным условием для корректной работы системы протоколирования. Планировщик `cron` необходимо настроить на регулярный запуск утилиты `logrotate`, т.к. программа-инсталлятор ПО ViPNet Coordinator Linux не конфигурирует планировщик.

При установке ПО ViPNet Coordinator Linux в каталоге `/etc/logrotate.d` создается файл `vipnet.log.conf`, в котором задаются параметры ротации файлов в каталоге `/var/log`: `iplircfg.debug.log`, `failover.debug.log`, `mftp.debug.log`, `alg.debug.log`. По умолчанию в файле `vipnet.log.conf` установлены следующие значения параметров:

- выполнять ротацию ежедневно;
- хранить журналы за последние 7 дней;
- сжимать старые журналы;
- после ротации и перед сжатием (опция `postrotate`) вызывать соответствующий скрипт с параметром `logrotate`:
  - для `iplircfg.debug.log` — скрипт `iplir`;
  - для `failover.debug.log` — скрипт `failover`;
  - для `mftp.debug.log` — скрипт `mftp`;
  - для `alg.debug.log` — скрипт `alg`.

Внимание! В файле конфигурации `/etc/logrotate.d/vipnet.log.conf` используются абсолютные пути к файлам журналов демонов ViPNet. Не рекомендуется менять эти пути. Однако если все же решено изменить параметр `debuglogfile`, необходимо также изменить его в файле конфигурации `/etc/logrotate.d/vipnet.log.conf`. Кроме того, нельзя переносить файлы журналов в другое место путем их замены на символические ссылки.

A

Процессы-демоны, входящие в состав ПО ViPNet Linux

При работе ViPNet Coordinator Linux в памяти постоянно находятся следующие программы-демоны:

- iplircfg — основной управляющий демон ViPNet, реализующий функции сервера IP-адресов, ведения журнала и так далее. Запускается командой `iplir start` и останавливается командой `iplir stop`.
- mftpd — демон MFTP, осуществляющий обработку и пересылку конвертов Деловой почты, прием обновлений справочников и ключей и так далее. Запускается командой `mftp start` и останавливается командой `mftp stop`.
- failoverd — демон системы защиты от сбоев (см. «Система защиты от сбоев» на стр. 126). Запускается командой `failover start` и останавливается командой `failover stop`.
- algd — демон обработки прикладных протоколов (см. «Настройка параметров обработки прикладных протоколов» на стр. 119). Запускается командой `alg start` и останавливается командой `alg stop`.
- axis2.cgi — демон, обеспечивающий функциональность сервера веб-интерфейса.

Все программы-демоны, входящие в состав ПО ViPNet Coordinator Linux, функционируют как процессы операционной системы, которые можно увидеть, выполнив, например, команды `ps aux` или `top`. При этом название соответствующего процесса отражает его текущее состояние, то есть функции, которые он выполняет в данный момент. В начале заголовка процесса пишется имя

демона (`iplircfg`, `mftpd`, `failoverd`, `algd`, `axis2.cgi`), а затем строка, идентифицирующая текущее состояние.

Ниже приведен список возможных состояний для процессов-демонов ПО ViPNet Coordinator Linux с кратким описанием.

Управляющий демон iplircfg:

- `initializing` – означает, что производится процедура инициализации демона, которая выполняется до начала выполнения его основных функций. Обычно процедура инициализации включает в себя чтение и анализ конфигурации, загрузку информации в драйвер и тому подобное и занимает короткое время (от одной до нескольких десятков секунд).
- `working` – означает, что процедура инициализации завершилась успешно и демон вошел в режим выполнения своих основных функций. Данное состояние остается до завершения работы процесса управляющего демона.

  В случае работы системы защиты от сбоев в режиме кластера горячего резервирования при запуске управляющего демона в пассивном режиме в конце заголовка процесса добавляется обозначение режима: `(passive)`.

Демон mftpd:

- `initializing` – означает, что производится процедура инициализации демона, которая выполняется до начала выполнения его основных функций. Обычно процедура инициализации занимает короткое время (от одной до нескольких десятков секунд).
- `updating` – означает, что запущен процесс удаленного обновления ключей, справочников или ПО.
- `transferring` – означает, что в данный момент производится активная передача конверта по одному из MFTP-каналов (кроме SMTP/POP3) как в случае приема, так и в случае передачи данных.
- `connecting` – означает, что в данный момент производятся попытки соединиться с одним из удаленных узлов (кроме канала SMTP/POP3) либо с демоном mftpd на пассивной части (в случае работы в составе кластера горячего резервирования).
- `idle` – означает, что демон mftpd в данный момент находится в режиме простоя, т.е. выполняет неосновные функции, не относящиеся ни к одному из вышеперечисленных состояний.

В любом состоянии к заголовку процесса mftpd также добавляется режим работы системы защиты от сбоев:

- `(single)` – одиночный режим работы;
- `(active)` – активный режим работы в составе кластера горячего резервирования;
- `(passive)` – пассивный режим работы в составе кластера горячего резервирования.

Демон системы защиты от сбоев failoverd:

- `initializing` – означает, что производится процедура инициализации демона, которая выполняется до начала выполнения его основных функций. Обычно процедура инициализации включает в себя чтение и анализ конфигурации, конфигурирование параметров сетевых интерфейсов, перезапуск других демонов и тому подобное, в зависимости от режима работы системы защиты от сбоев. Данная процедура занимает короткое время (от одной до нескольких десятков секунд).
- `working` – означает, что процедура инициализации завершилась успешно и демон вошел в режим выполнения своих основных функций.
- `switching to active` – означает, что производится настройка системы при переключении из пассивного режима в активный. Данное состояние возможно только в случае, если система защиты от сбоев функционирует в режиме кластера горячего резервирования.
- `rebooting` – означает, что демоном failoverd инициирован процесс перезагрузки системы.

В любом состоянии (кроме `switching to active`) к заголовку процесса добавляется режим работы системы защиты от сбоев:

- `(single)` – одиночный режим работы;
- `(cluster) (active)` – активный режим работы в составе кластера горячего резервирования;
- `(cluster) (passive)` – пассивный режим работы в составе кластера горячего резервирования.

Демон обработки прикладных протоколов algd:

- `initializing` – означает, что производится процедура инициализации демона, которая выполняется до начала выполнения его основных функций. Обычно процедура инициализации занимает короткое время (от одной до нескольких десятков секунд).
- `working` – означает, что процедура инициализации завершилась успешно и демон вошел в режим выполнения своих основных функций. Данное состояние остается до завершения работы процесса управляющего демона.

B

Модули ядра, входящие в состав ПО ViPNet Linux

При работе ViPNet Coordinator Linux в памяти постоянно находятся следующие модули ядра системы (также называемые драйверами):

- **drviplir** — основной драйвер ViPNet, осуществляющий обработку сетевых пакетов. Порождает потоки по количеству ядер CPU (по умолчанию) или по количеству, заданному параметром `threads_count` (см. «Управление числом потоков в драйвере» на стр. 155), которые видны в списке процессов как `[kiplird0]`, `[kiplird1]` и так далее, при выгрузке уничтожает их. Загружается по команде `iplir start`, выгружается по команде `iplir unload`. Не может быть выгружен вручную командой `rmmod`.
- **itcscrpt** — криптографический драйвер, осуществляющий криптографические операции по запросу драйвера drviplir. Загружается по команде `iplir start`, выгружается по команде `iplir unload`.
- **itcswd** — драйвер системы защиты от сбоев, осуществляющий контроль за состоянием системы. Порождает поток ядра, который виден в списке процессов как `[kwatchdogd]`, при выгрузке уничтожает его. Загружается по команде `vipmod start`, после остановки демона failoverd и выгрузки драйвера drviplir может быть выгружен вручную командой `rmmod`.
- **itcskrniface** — драйвер, который является интерфейсом экспортируемых функций ядра и учитывает различия этих функций для разных версий ядер.

Действия по выгрузке и загрузке драйверов ViPNet вручную должен производить только опытный администратор Linux.

C

Изменения, производимые в системе при установке и удалении ViPNet Coordinator Linux

Дистрибутив ViPNet Coordinator Linux может поставляться в двух вариантах: динамическом и статическом. В динамическом варианте общие части кода различных программ поставляются в виде динамически загружаемых библиотек. При этом экономится место на диске и используемая оперативная память, однако динамически загружаемые библиотеки могут не работать при нестандартных настройках системы. В статическом дистрибутиве все программы независимы и не требуют никаких динамических библиотек. Статический дистрибутив отличается наличием суффикса `-static` в имени файла. В дальнейшем в этом приложении, если иное не указано явно, речь идет о поведении обоих вариантов дистрибутивов.

При установке:

1 В каталог `/sbin` записываются файлы:

```
iplircfg
iplircfg.crg
dbviewer
dbviewer.crg
iplir_remote_info
iplir_remote_info.crg
```

```
iplir-config-manager
iplir-config-manager.crg
failoverd
failoverd.crg
failover_info
failoverd_info.crg
unmerge
unmerge.crg
fetchmail
fetchmail.crg
iplirpasswd
iplirpasswd.crg
mftpd
mftpd.crg
algd
algd.crg
mailtrans
mailtrans.crg
mftp_remote_info
mftp_remote_info.crg
iplir_monitoring_export
iplir_monitoring_export.crg
monitoringcfg
monitoringcfg.crg
webgui-http-server
webgui-http-server.crg
iplir
vipmod
failover
mftp
rmiplir
alg
vipnet-web-gui
```

2 В каталог `/usr/bin` записываются файлы:

```
lha
lha.crg
```

3 В каталог `/etc` записываются файлы:

```
iplirprg
iplirprg.crg
```

iplirpsw

iplirnetpsw

failover.ini

iplir.serv

iplirSKnums

/etc/logrotate.d/vipnet.log.conf

4 Создается каталог /var/spool/vipnet

5 Ищутся сначала в /etc/rc.d, если не найдены - то в /etc, каталоги:

init.d

rc0.d

rc1.d

rc2.d

rc3.d

rc4.d

rc5.d

rc6.d

- В каталог init.d ставятся линки (symbolic links):

  vipmod -> /sbin/vipmod

  failover -> /sbin/failover

- В каталоги rc0.d, rc1.d, rc2.d ставятся линки:

  K<XX>failover -> /etc/init.d/failover

  K<YY>vipmod -> /etc/init.d/vipmod

  где XX, YY могут меняться от версии к версии.

- В каталоги rc2.d, rc3.d, rc4.d, rc5.d ставятся линки:

  S<XX>vipmod -> /etc/init.d/vipmod

  S<YY>failover -> /etc/init.d/failover

  где XX, YY могут меняться от версии к версии.

6 В каталог /lib/modules/<версия_ядра>/misc записываются файлы:

drviplir.o (drviplir.ko для ядер серии 2.6.x)

itcswd.o (itcswd.ko для ядер серии 2.6.x)

itcscrpt.o (itcscrpt.ko для ядер серии 2.6.x)

где версия_ядра определяется как результат выполнения команды uname -r.

7 Если дистрибутив динамический, то создается каталог /usr/lib/vipnet и в него записываются файлы библиотек, список которых может меняться от версии к версии.

При установке поверх предыдущей версии не перезаписываются файлы в каталоге /etc (кроме случаев установки без сохранения текущей конфигурации ViPNet):

```
iplirpsw
iplirnetpsw
failover.ini
```

Все остальные файлы перезаписываются. Если статический дистрибутив ставится поверх динамического, то каталог `/usr/lib/vipnet` не удаляется, хотя и перестает использоваться.

При удалении ПО ViPNet из системы удаляются все перечисленные файлы и линки (в том числе `/usr/lib/vipnet`, даже если удаление производится из статической версии), кроме файла `/sbin/rmiplir`.

D

Особенности работы ViPNet Coordinator Linux на различных аппаратных конфигурациях

В этом приложении собрана информация об особенностях работы ViPNet Coordinator Linux на специфических аппаратных конфигурациях (например, на некоторых брендовых серверах) и о настройках, которые необходимо сделать в каждом случае.

Серверы Supermicro: SuperServer 5013G-i (SYS-5013-Gi)

Для успешной работы ViPNet Coordinator Linux на этих серверах необходимо отключать поддержку Advanced Power Management (APM). Это можно сделать либо при сборке ядра, выключив соответствующую опцию, либо при загрузке, задав параметр `apm=off`. Если управление питанием все же необходимо, нужно использовать ACPI. Этот интерфейс предоставляет все возможности, которые предоставляет APM, и нормально работает на таких серверах.

Сетевые Ethernet-адаптеры с поддержкой технологий «Scatter/Gather I/O» и «TSO»

Во многих современных Ethernet-адаптерах (например, Intel 82541 и другие) для повышения скорости передачи сетевых пакетов по умолчанию используются технологии «Scatter/Gather I/O» и «TSO (TCP Segmentation Offload)», позволяющие оптимизировать процесс передачи/приема пакетов аппаратными средствами адаптера.

Внимание! Данные технологии не совместимы с работой текущей версии драйвера сетевой защиты iplir, и должны быть отключены для корректной работы ПО ViPNet.

Отключить поддержку описанных механизмов можно, например, с помощью широко распространенной утилиты `ethtool`:

```
# ethtool -K eth<N> sg off
# ethtool -K eth<N> tso off
```

где `N` – номер сетевого интерфейса.

Проверить текущее состояние «Scatter/Gather I/O» и «TSO» можно следующей командой:

```
# ethtool -k eth<N>
```

где `N` – номер сетевого интерфейса.

В случае если данные механизмы отключены, команда выдает на стандартный вывод следующее сообщение:

```
scatter-gather: off
tcp segmentation offload: off
```

За более подробным описанием утилиты `ethtool`, поставляемой с используемым дистрибутивом Linux, следует обратиться к документации (`man ethtool`).

Примечание. В процессе старта ПО ViPNet при загрузке ОС указанные технологии оптимизации автоматически отключаются, о чем выводится предупреждение в `syslog`.

E

События, отслеживаемые ПО ViPNet Coordinator Linux

События, отслеживаемые ПО ViPNet Coordinator Linux, разделены на группы и подгруппы.

Рисунок 20. Классификация событий в журнале IP-пакетов

Нижеследующие таблицы содержат списки событий, относящихся к разным группам и подгруппам.

Таблица 12. События подгруппы ***Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные сетевыми фильтрами защищенной сети***

Номер события	Название события	Описание события
1	Не найден ключ для сетевого узла	Не найден ключ для связи с пользователем, идентификатор которого указан в пакете
2	Неверное значение имито	Защищаемые данные или открытая информация криптосистемы были изменены
3	IP-пакет блокирован фильтром защищенной сети	Согласно настройкам фильтров, входящий зашифрованный или предназначенный для шифрования исходящий открытый пакет был заблокирован
4	Слишком большая разница во времени	Пакет был отправлен раньше или позже даты, установленной на принимающем узле, на величину большую, чем указано в настройке допустимого времени приема пакетов
5	Некорректная версия драйвера	Принят пакет, созданный версией драйвера, несовместимой с текущей версией
7	Неизвестный метод шифрования	Не поддерживается метод шифрования, код которого указан во входящем пакете
8	Искаженный IP-пакет	Недопустимые параметры в расшифрованном пакете
9	Неизвестный идентификатор сетевого узла	Идентификатор отправителя в пакете неизвестен
10	IP-пакет блокирован главным фильтром защищенной сети	Пакет блокирован фильтром, относящимся к записи «IP-пакеты всех адресатов»
13	Превышено время жизни IP-пакета	Пакет уничтожен из-за превышения лимита его нахождения в сети
14	Получен IP-пакет для другого сетевого узла	Принят пакет для другого адресата
15	Слишком много фрагментов для IP-пакета	Превышено допустимое количество одновременно обрабатываемых фрагментированных пакетов
16	Исчерпана лицензия на количество туннелируемых адресов	На координатор, осуществляющий туннелирование, одновременно пришли пакеты от компьютеров, число которых превышает число заданных в лицензии туннелей
17	Неверный IP-адрес	В пришедшем зашифрованном пакете Ethernet свой, а IP-адрес чужой, и при этом пакет не является маршрутизируемым (событие 45)
18	Неизвестный IP-адрес получателя	Событие появляется в случае, если координатор не знает, на какой адрес перенаправить входящий пакет

Номер события	Название события	Описание события
70	Транзитный IP-пакет блокирован фильтром защищенной сети	Транзитный зашифрованный IP-пакет заблокирован на координаторе фильтром защищенной сети

*Таблица 13. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные сетевыми фильтрами открытой сети***

Номер события	Название события	Описание события
22	Незашифрованный IP-пакет от сетевого узла	От защищенного адресата пришел открытый пакет
23	Незашифрованный широковещательный IP-пакет от сетевого узла	От защищенного адресата пришел открытый широковещательный пакет
30	Локальный IP-пакет блокирован фильтром открытой сети	Найден запрещающий фильтр открытой сети в группе локальных фильтров
31	Транзитный IP-пакет блокирован фильтром открытой сети	Найден запрещающий фильтр открытой сети в группе транзитных фильтров
32	Широковещательный IP-пакет блокирован фильтром открытой сети	Найден запрещающий фильтр открытой сети в группе широковещательных фильтров
37	Пакет блокирован фильтром для туннелируемых ресурсов	Найден запрещающий фильтр открытой сети в группе правил для туннелируемых ресурсов
39	IP-пакет блокирован фильтрами по умолчанию при загрузке компьютера	При загрузке компьютера в фильтрах по умолчанию найден запрещающий фильтр открытой сети

*Таблица 14. События подгруппы **Все IP-пакеты/Блокированные IP-пакеты/IP-пакеты, блокированные по другим причинам***

Номер события	Название события	Описание события
80	Размер IP-пакета меньше допустимого	Размер IP-пакета меньше минимально возможного
81	Недопустимая версия протокола IP	В данной версии поддерживается только протокол IP версии 4

Номер события	Название события	Описание события
82	Недопустимая длина заголовка IP	Длина заголовка протокола IP меньше минимально возможного
83	Недопустимая длина IP-пакета	Длина пакета меньше, чем указано в заголовке протокола IP
84	Несовпадение контрольной суммы IP	Подсчитанное значение контрольной суммы заголовка IP-пакета не совпадает со значением, указанным в пакете
85	Размер заголовка TCP меньше минимально допустимого	Недопустимо короткий заголовок протокола TCP
86	Размер заголовка UDP меньше минимально допустимого	Недопустимо короткий заголовок протокола UDP
87	Дефрагментация IP-пакета была отменена	Были обработаны не все фрагменты, образующие пакет
88	Широковещательный адрес отправителя IP-пакета	Адрес отправителя в пакете указан широковещательный
89	Фрагменты IP-пакета пересекаются между собой	Наиболее старый из пересекающихся фрагментов был отброшен
90	Недостаточно ресурсов для обработки IP-фрагмента	Пакет не может быть обработан из-за недостаточности свободных ресурсов Если эта ошибка стабильно проявляется, то требуется обновление версии драйвера, использующего больше машинных ресурсов, или требуется более совершенная модель компьютера
91	IP-пакет получен во время инициализации драйвера	Блокировка всех пакетов во время инициализации драйвера
92	Слишком большой размер IP-пакета	Размер пакета ограничен размером 48 Кбайт
93	Превышено время сборки фрагментов IP-пакета	За допустимое время получены не все фрагменты IP-пакета
94	Недостаточно памяти	Для выполнения какой-либо операции требуется выделение памяти, но выделить память не удалось
95	Обнаружен сетевой узел с таким же идентификатором	Поступили пакеты с одинаковыми идентификаторами узлов, но разными IP-адресами
98	Недостаточно ресурсов для обработки IP-пакета	Для обработки IP-пакета требуется выделение ресурсов, но выделить ресурсы не удалось
100	Недопустимые флаги TCP	Это событие не используется
101	Не найден маршрут для транзитного IP-пакета	Не найден сетевой фильтр для транзитного пакета в таблице маршрутов

Номер события	Название события	Описание события
102	Модуль прикладной обработки не загружен	Не загружен соответствующий модуль прикладной обработки
103	Превышено максимальное количество соединений	Количество уже установленных соединений превышает максимально допустимое, заданное в настройках координатора
104	Соединение уже существует	Для создаваемого соединения параметры исходящих пакетов (socketpair) совпадают с уже существующими, такое соединение блокируется
105	Не удалось выделить динамический порт для правила трансляции адресов	Не удалось выделить порт для динамической трансляции адресов (например, все порты в пуле закончились)
115	Не удалось найти маршрут для IP-пакета	По каким-либо причинам не найден маршрут в таблице маршрутизации
116	Сетевой адаптер не найден	IP-пакет не может быть отправлен, так как не найден сетевой адаптер
117	Не удалось разрешить MAC-адрес по IP-адресу	Не удалось определить MAC-адрес получателя пакета по его IP-адресу
118	Не удалось произвести шифрование IP-пакета	Ошибка при шифровании исходящего IP-пакета для защищенного узла
119	Неизвестный формат IPLIR заголовка	Получен зашифрованный IP-пакет неизвестного формата
120	Несогласованная информация о способе доступа до сетевого узла	Ошибка при отправке IP-пакета для защищенного узла
121	Ошибка в работе кластера	Это событие регистрируется только на кластере ViPNet. Внутренняя ошибка кластера
122	Неизвестный протокол канального уровня	Получен IP-пакет неизвестного протокола

*Таблица 15. События группы **Все IP-пакеты/Все пропущенные IP-пакеты/Пропущенные зашифрованные IP-пакеты***

Номер события	Название события	Описание события
40	Пропущен зашифрованный IP-пакет	Разрешенный зашифрованный пакет
41	Пропущен зашифрованный широковещательный IP-пакет	Разрешенный зашифрованный широковещательный пакет

Номер события	Название события	Описание события
44	Осуществлена маршрутизация зашифрованного транзитного IP-пакета с изменением его адреса	Пакет направлен на другой узел путем подмены в нем адреса получателя
45	Зашифрован (расшифрован) пакет туннелируемого ресурса	Это событие для координатора, осуществляющего туннелирование: пакет пропущен, так как поступил от туннелируемого узла или предназначен для него

*Таблица 16. События группы **Все IP-пакеты/Все пропущенные IP-пакеты/Пропущенные незашифрованные IP-пакеты***

Номер события	Название события	Описание события
60	Пропущен незашифрованный локальный IP-пакет	Найден разрешающий сетевой фильтр открытой сети в группе локальных фильтров
61	Пропущен незашифрованный широковещательный IP-пакет	Найден разрешающий фильтр открытой сети в группе широковещательных фильтров
62	Пропущен незашифрованный транзитный IP-пакет	Найден разрешающий фильтр открытой сети в группе транзитных фильтров
63	Пакет пропущен фильтром для туннелируемых ресурсов	Найден разрешающий фильтр в группе фильтров для туннелируемых узлов

*Таблица 17. События группы **Все IP-пакеты/Служебные события***

Номер события	Название события	Описание события
42	Изменился адрес сетевого узла	Драйвер обнаружил, что IP-адрес сетевого узла изменился и соответствующим образом скорректировал свои таблицы
46	Изменился адрес доступа к сетевому узлу	Событие предназначено для диагностики настройки правил NAT на межсетевых экранах, через которые работают другие узлы. Это событие может появиться, если данный узел не установлен в режим работы через координатор Событие формируется об узлах, установленных в режим работы через межсетевой экран, в том случае, если на межсетевом экране изменился адрес или порт доступа к этим узлам со стороны данного узла

Номер события	Название события	Описание события
48	Адрес сетевого узла зарегистрирован из широковещательного пакета	Технологический код Формируется для удаленного узла, если этот узел становится доступным или перестает быть доступным по широковещательным пакетам
114	Имя на DNS (WINS)-сервере не зарегистрировано	Поступило сообщение от DNS-сервера, что запрошенное DNS-имя защищенного узла не зарегистрировано на данном DNS-сервере

F

Особенности работы координатора на границе локальной сети

В данном приложении описаны особенности работы координаторов с ПО ViPNet Coordinator Linux, расположенных на границе локальной сети и Интернета. Рассмотрим схему взаимодействия двух таких координаторов.

Рисунок 21: Определение адреса доступа координатора

Пусть у каждого координатора есть два сетевых интерфейса. При этом один сетевой интерфейс имеет публичный адрес и подключен к Интернету (внешний интерфейс `eth0`), а второй — частный адрес и подключен к локальной сети (внутренний интерфейс `eth1`). Для определения работоспособности друг друга координаторы периодически обмениваются служебными пакетами (подробнее см. описание параметров `server_pollinterval` и `client_pollinterval` в разделе Секция [misc] (на стр. 58)). Кроме того, каждый раз при запуске, а также в случае восстановления связи после ее потери координатор 1 отправляет через интерфейс `eth0` тестовые UDP-пакеты на публичный и частный адреса координатора 2 (по умолчанию порт назначения — 55777) для определения его доступности. Аналогичным образом UDP-пакеты могут следовать с интерфейса

`eth0` координатора 2 на публичный и частный адреса координатора 1. Таким образом, UDP-пакеты, направленные на частные IP-адреса, могут появиться в Интернете.

Внимание! Появление в Интернете UDP-пакетов, направленных на частные IP-адреса, может вызвать вопросы у провайдера, предоставляющего вам интернет-услуги. Во избежание этого рекомендуется на обоих координаторах настроить дополнительные правила маршрутизации таких UDP-пакетов не через Интернет (то есть не через шлюз по умолчанию). Причем и на координаторе 1, и на координате 2 рекомендуется настроить правила маршрутизации UDP-пакетов для обеих локальных сетей: 10.0.1.1 и 10.0.2.1.

G

Практические примеры настройки ПО ViPNet Coordinator Linux

Примеры настройки туннелей с использованием координаторов ViPNet

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 координатор ViPNet Coordinator Linux, к которому подключаются компьютеры (см. схему).

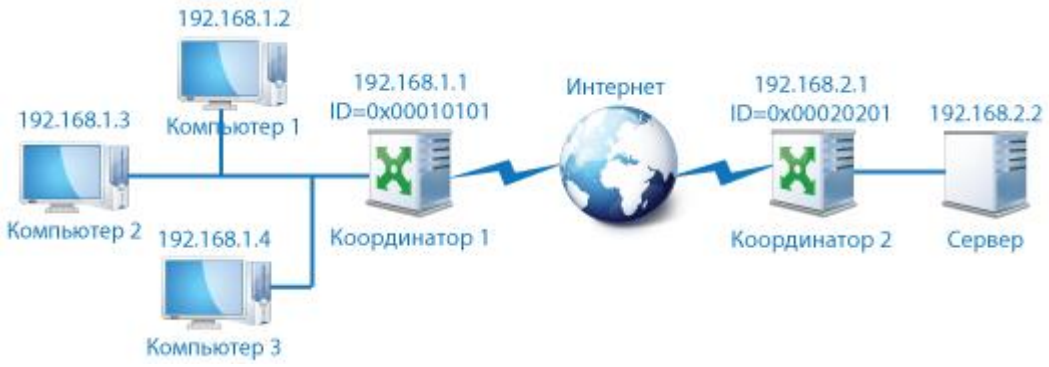

Рисунок 22. Схема настройки туннелей с использованием координаторов ViPNet

Каждый координатор имеет два сетевых интерфейса, один из которых имеет публичный адрес и подключен к Интернету (внешний интерфейс), а второй имеет частный адрес и подключен к локальной сети офиса (внутренний интерфейс). Пусть Координатор 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`. Координатор 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 координаторах необходимо сделать следующие настройки в файле `iplir.conf`:

- На Координаторе 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 Координатора 2 вписать строку:

    `tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2`
- На Координаторе 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 Координатора 1 вписать строку:

    `tunnel= 192.168.1.2-192.168.1.4 to 192.168.1.2-192.168.1.4`

После запуска управляющего демона с указанными настройками компьютеры 1, 2, 3 смогут обращаться к серверу по адресу `192.168.2.2`. При этом на участке между Координатором 1 и Координатором 2 пакеты будут идти в зашифрованном виде.

По умолчанию на координаторах настроены сетевые фильтры, разрешающие трафик для всех туннелируемых адресов. Если в приведенном примере требуется закрыть доступ к серверу каким-либо компьютерам, необходимо изменить сетевые фильтры с помощью командного интерпретатора 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 Координаторе 1 необходимо удалить сетевой фильтр по умолчанию и вместо него создать следующий фильтр:

```
firewall tunnel add src 192.168.1.3-192.168.1.4 dst 0x00020201 pass
```

Этот сетевой фильтр разрешает трафик в случаях когда инициаторами соединения являются клиенты, а не сервер. В свою очередь, на Координаторе 2 должен быть разрешен трафик между сервером и Координатором 1. Для этого на Координаторе 2 можно оставить сетевой фильтр по умолчанию или вместо него задать фильтр, явно разрешающий трафик между туннелируемым сервером и Координатором 1:

```
firewall tunnel add src 0x00010101 dst 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 Координаторе 1 необходимо задать следующий фильтр:

```
firewall tunnel add src 192.168.1.3-192.168.1.4 dst 0x00020201 tcp dport 80 pass
```

На Координаторе 2 необходимо задать следующий сетевой фильтр:

```
firewall tunnel add src 0x00010101 dst 192.168.2.2 tcp dport 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 23. Модифицированная схема настройки туннелей

В этом случае, когда туннелируется только сервер, на координаторах ViPNet необходимо сделать следующие настройки в файле `iplir.conf`:

- На Координаторе 1:
  - В секции `[id]` для Координатора 2 вписать строку:

    `tunnel= 192.168.2.2-192.168.2.2 to 192.168.2.2-192.168.2.2`
- На Координаторе 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 Координатора 2 задан сетевой фильтр по умолчанию для туннелируемых адресов, то никакие дополнительные настройки не нужны. Иначе на Координаторе 2 необходимо задать фильтр, разрешающий трафик между сервером и компьютерами 1, 2, 3:

```
firewall tunnel add src 0x00010102,0x00010103,0x00010104 dst 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 Координаторе 2 необходимо изменить сетевой фильтр следующим образом:

```
firewall tunnel add src 0x00010103,0x00010104 dst 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, приведенный фильтр необходимо изменить следующим образом:

```
firewall tunnel add src 0x00010102,0x00010103,0x00010104 dst 192.168.2.2 tcp dport
80 pass
```

Пример использования дополнительных IP-адресов на интерфейсе

В данном приложении приведен пример использования дополнительных IP-адресов на одном из интерфейсов координатора ViPNet Coordinator Linux. Способ задания дополнительных адресов, описанный в примере, справедлив только для одиночного координатора. Если координатор входит в состав кластера горячего резервирования, то дополнительные адреса задаются с помощью специальных параметров настройки кластера. Способ задания дополнительных адресов на кластере горячего резервирования приведен в документе «ViPNet Coordinator Linux. Система защиты от сбоев. Руководство администратора».

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 координатор ViPNet Coordinator Linux, к которому подключаются серверы (см. схему).

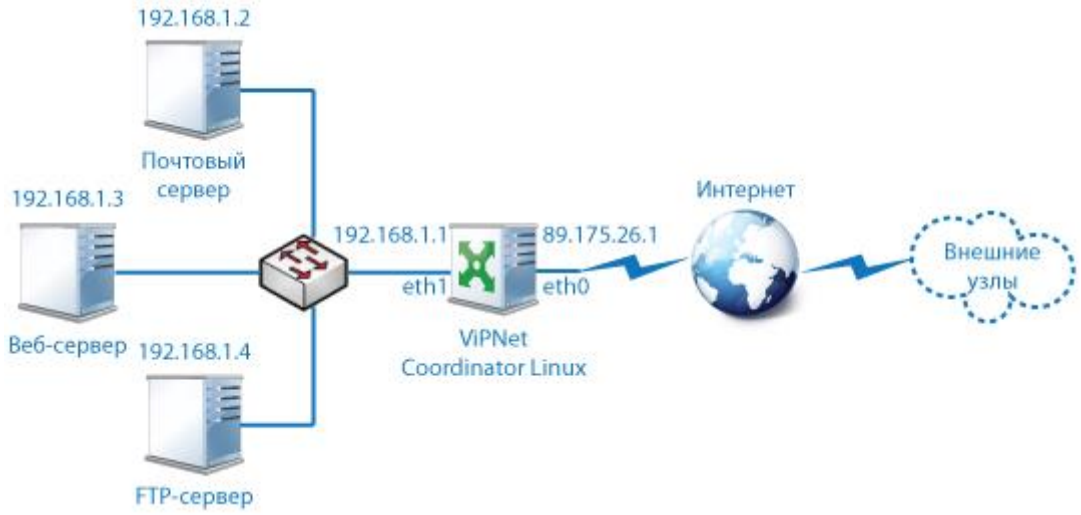

Рисунок 24. Схема подключения серверов к координатору

Координатор 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 координатор 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 координатор работает в режиме **Со статической трансляцией адресов**. Чтобы обеспечить доступ к серверам извне, на координаторе необходимо выполнить следующее:

- Задать дополнительные адреса на внешнем интерфейсе `eth0` с помощью следующих команд:

```
ifconfig eth0:0 89.175.26.2 netmask 255.255.255.0
ifconfig eth0:1 89.175.26.3 netmask 255.255.255.0
ifconfig eth0:2 89.175.26.4 netmask 255.255.255.0
```

- В командном интерпретаторе задать следующие правила трансляции адреса получателя:

```
firewall nat add src @any dst 89.175.26.2 tcp dport 80 change dst
192.168.1.2:25
firewall nat add src @any dst 89.175.26.3 tcp dport 81 change dst
192.168.1.3:80
firewall nat add src @any dst 89.175.26.4 tcp dport 82 change dst
192.168.1.3:21
```

- В командном интерпретаторе задать следующие разрешающие сетевые фильтры:

```
firewall forward add src @any dst 192.168.1.2 tcp dport 25 pass
firewall forward add src @any dst 192.168.1.3 tcp dport 80 pass
firewall forward add src @any dst 192.168.1.4 tcp dport 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 координатора.

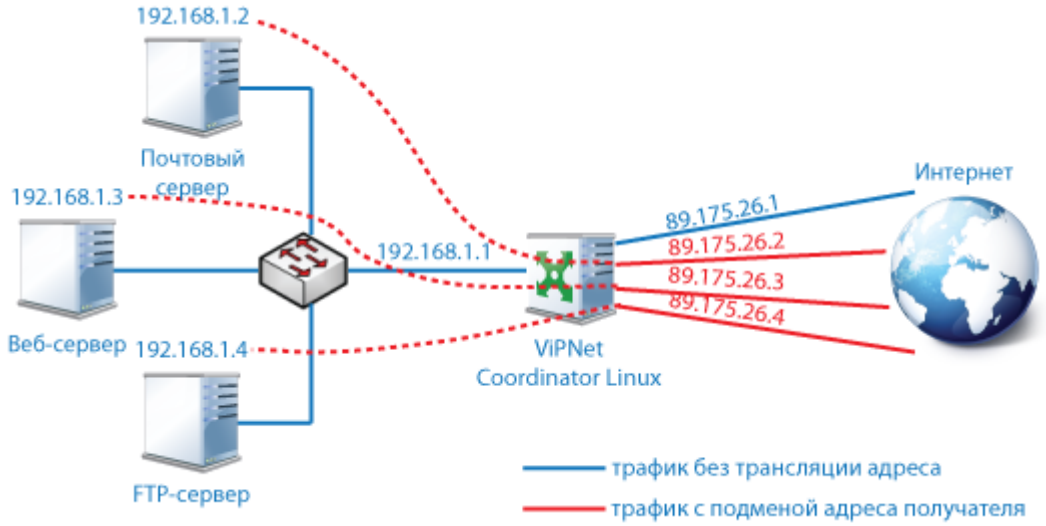

Рисунок 25. Схема организации доступа к серверам с помощью дополнительных адресов

H

Справочник команд

Примечание. Команды, которые можно выполнить только в режиме администратора, отмечены красным цветом.

Команды группы firewall

Основные команды для работы с сетевыми фильтрами, правилами трансляции IP-адресов и группами объектов:

- `firewall <тип сетевого фильтра или правила> show` — просмотр сетевых фильтров определенного типа или правил трансляции адресов, используемых в настройках своего узла.
- `firewall <тип сетевого фильтра или правила> add <номер фильтра> rule <название фильтра> <условие> <расписание> <действие>` — добавление сетевых фильтров или правил трансляции адресов.
- `firewall <тип сетевого фильтра или правила> change append <номер сетевого фильтра или правила> <дополнительное условие>` — изменение параметров сетевого фильтра или правила трансляции адресов.
- `firewall <тип сетевого фильтра или правила> delete <номер сетевого фильтра или правила>` — удаление сетевого фильтра или правила трансляции адресов.
- `firewall <тип сетевого фильтра или правила> move rule <старый номер фильтра или правила> to <новый номер фильтра или правила>` — изменение порядкового номера фильтра в таблице. Если задан новый номер, превышающий общее количество фильтров или правил в списке, то выдается сообщение об этом, а фильтр или правило не перемещается.
- `firewall <тип группы объектов> show` — просмотр групп объектов определенного типа.

- firewall <тип группы> add name <название группы> <содержимое группы> [exclude <исключения группы>] — добавление группы объектов.
- firewall <тип группы> change append <название группы> {include | exclude} <объект или несколько объектов> — добавление или удаление из существующей группы каких-либо объектов
- firewall object delete <название группы> — удаление группы объектов.

Команды группы iplir option

Команды группы iplir option предназначены для управления функцией антиспуфинга (см. «Антиспуфинг» на стр. 107), а также для тонкой настройки дополнительных опций межсетевого экрана.

- iplir option set antispoofing {on | off} — включение или отключение антиспуфинга. По умолчанию антиспуфинг выключен.
- iplir option set block-other-protocols {on | off} — включение или отключение блокировки передающихся не по протоколам IP, ARP, RARP. По умолчанию блокировка выключена.
- iplir option set max-connections <значение> — задание максимального количества одновременных соединений. Значение по умолчанию — 150000.
- iplir option set dynamic-ports <порт>-<порт> — задание диапазона портов для динамической трансляции адресов. Значение по умолчанию — 60000-65000.
- iplir option set connection-ttl-tcp <значение> — задание тайм-аута соединений по протоколу TCP. Значение по умолчанию — 3600 секунд.
- iplir option set connection-ttl-udp <значение> — задание тайм-аута соединений по протоколу UDP. Значение по умолчанию — 300 секунд.
- iplir option set connection-ttl-ip <значение> — задание тайм-аута соединений по протоколу IP. Значение по умолчанию — 60 секунд.
- iplir option set dynamic-timeouts {yes | no} — включение или отключение функции динамических тайм-аутов. По умолчанию функция отключена.
- iplir option set cleanup-interval <значение> — задание интервала удаления устаревших соединений. Значение по умолчанию — 5 секунд.
- iplir option get <название опции> — просмотр состояния или значения одной из опций (в том числе антиспуфинга).

Команды группы alg

Команды группы `alg` (Application Layer Gateway) предназначены для работы с параметрами обработки прикладных протоколов (см. «Настройка параметров обработки прикладных протоколов» на стр. 123):

- `alg show` — просмотр текущих параметров обработки прикладных протоколов.
- `alg module <имя прикладного протокола> process <название сетевого протокола для обработки> <порт или диапазон портов для обработки> on` — включение или изменение настройки обработки прикладных протоколов.

  Поддерживаемые прикладные протоколы — `sip`, `h323`, `ftp`, `sccp`, `dns`.

  Сетевые протоколы для обработки — `tcp`, `udp`.

  Если для какого-либо прикладного протокола будет установлено значение порта равным 0, то обработка протокола будет выключена.
- `alg module <ModuleName> process off` — отключение обработки прикладного протокола.
- `alg start` — запуск демона обработки прикладных протоколов algd.
- `alg stop` — остановка демона algd.

Примечание. Для обеспечения бесперебойной работы прикладных сервисов в процессе штатного функционирования ПО ViPNet Coordinator Linux не рекомендуется останавливать демон algd.

I

История версий

Что нового в версии 4.1.3

В этом разделе представлен краткий обзор изменений и новых возможностей ViPNet Coordinator Linux версии 4.1.3.

- **Дополнительные настройки туннелирования**

  Раньше настройки туннелирования на координаторе ограничивались заданием диапазона IP-адресов туннелируемых незащищенных узлов. Теперь в случае необходимости вы можете исключить один или несколько адресов из диапазона туннелирования, а также включить или выключить туннелирование незащищенных компьютеров с помощью специального параметра (см. «Секция [id]» на стр. 49).

  Кроме того, раньше соединение ViPNet Coordinator Linux с туннелируемыми узлами, находящимися с ним в одной локальной подсети, всегда осуществлялось через координатор, который туннелирует данные узлы. По этой причине доступ к таким узлам мог быть затруднен. Теперь по умолчанию соединение ViPNet Coordinator Linux с туннелируемыми узлами, находящимися с ним в одной локальной подсети, осуществляется напрямую, минуя туннелирующий координатор. Для совместимости с более ранними версиями предусмострена возможность изменения этой настройки на ViPNet Coordinator Linux (см. «Секция [misc]» на стр. 58).

- **Новая пользовательская группа объектов, настроенная по умолчанию**

  Раньше для упрощения создания сетевых фильтров и правил трансляции в ПО ViPNet Coordinator Linux были предусмотрены две настроенные по умолчанию пользовательские группы IP-адресов: группа с частными IP-адресами и группа с публичными IP-адресами. Теперь по умолчанию также создана группа, в составе которой указан адрес прокси-сервера, поддерживающего протокол HTTP (см. «Пользовательские группы объектов, настроенные по

умолчанию» на стр. 87). С помощью этой группы вы можете настроить доступ пользователей корпоративной сети к различным интернет-ресурсам.

Что нового в версии 4.1.2

В ViPNet Coordinator Linux версии 4.1.2 исправлены ошибки, выявленные при эксплуатации версии 4.1.

Что нового в версии 4.1

В этом разделе представлен краткий обзор изменений и новых возможностей ViPNet Coordinator Linux 4.1.

- **Поддержка расписаний**

  Появилась возможность привязывать действие сетевых фильтров к определенным промежуткам времени — применять фильтры по расписанию (см. «Расписание» на стр. 96). Например, вы можете создать сетевой фильтр, который будет блокировать доступ к некоторым веб-ресурсам в рабочее время в будние дни.

- **Работа с группами объектов**

  Теперь вы можете объединять однотипные объекты в именованные группы (см. «Использование групп объектов» на стр. 85). Группы объектов облегчают ввод условий при создании сетевых фильтров и правил трансляции, если в них требуется указать целый ряд объектов одного типа. Например, если требуется создать несколько сетевых фильтров, в которых нужно указать IP-адреса сегмента сети, вы можете предварительно создать группу этих IP-адресов и далее добавлять ее в условия фильтров.

- **Веб-интерфейс для работы с межсетевым экраном и списком узлов защищенной сети**

  В состав ПО ViPNet Coordinator Linux 4.1 был добавлен удобный веб-интерфейс, с помощью которого вы можете настраивать сетевые фильтры, правила трансляции IP-адресов, а также работать со списком узлов защищенной сети ViPNet. Подключение к веб-интерфейсу возможно с любого узла сети ViPNet, связанного с данным координатором. Подробнее о веб-интерфейсе см. в документе «ViPNet Coordinator Linux. Работа с веб-интерфейсом».

- **Настройка множественных адресов доступа узлов**

  Для корректной работы ПО ViPNet Coordinator Linux в качестве сервера IP-адресов появилась возможность настраивать множественные адреса доступа узлов, связанных с данным координатором (см. «Секция [id]» на стр. 49). При настройке адресов доступа какого-либо узла вы можете задать приоритеты данных адресов (какие из адресов являются более или менее предпочтительными для установления соединения). После выполнения настроек координатор с помощью межузловых рассылок информирует другие узлы о предпочтительных адресах доступа к какому-либо узлу.

- **Удален функционал фиксированных альтернативных каналов связи**

Теперь для узлов ViPNet можно указывать множественные адреса доступа, поэтому настраивать взаимодействие между узлами по фиксированным альтернативным каналам больше не требуется, данная функция удалена из ПО ViPNet Coordinator Linux. При обновлении ПО ViPNet Coordinator Linux до версии 4.1 и выше автоматическая конвертация настроек фиксированных альтернативных каналов не производится, поэтому их необходимо выполнить вручную (см. «Переход от использования фиксированных альтернативных каналов к множественным адресам доступа» на стр. 39).

- **Упрощение настройки режимов работы через межсетевой экран со статической и динамической трансляцией адресов**

  Координатор имеет несколько внешних сетевых интерфейсов, и у пользователей могут возникнуть трудности, какой из интерфейсов определить, как внешний. Также часто внешний интерфейс координатора, стоящий за межсетевым экраном с трансляцией адресов, получает адрес динамически, что делает невозможным централизованно (из ЦУСа) установить для координатора требуемый тип межсетевого экрана. В связи с этим была изменена логика настройки ViPNet Coordinator Linux. Теперь при настройке режима работы «Со статической трансляцией адресов» или «С динамической трансляцией адресов» больше не нужно для одного из сетевых интерфейсов координатора устанавливать тип `external` (за исключением режима статической трансляции адресов с фиксацией внешнего адреса).

- **Поддержка архитектуры ARM и новых дистрибутивов ОС Linux**

  Теперь возможна работа ПО ViPNet Coordinator Linux на дистрибутиве Astra Linux 1.4. Также теперь возможна работа ПО ViPNet Coordinator Linux на компьютерах с архитектурой процессора ARM со следующими дистрибутивами:

  - Debian 7 wheezy;
  - Ubuntu 12.04;
  - Picuntu 12.04.

Что нового в версии 4.0

В этом разделе представлен краткий обзор изменений и новых возможностей ViPNet Coordinator Linux 4.0.

- **Поддержка совместной работы с ПО ViPNet Policy Manager версии 4.0**

  Теперь ViPNet Coordinator Linux может принимать и обрабатывать политики безопасности, присланные из программы ViPNet Policy Manager версии 4.0.

- **Новый формат сетевых фильтров и правил трансляции IP-адресов**

  В ПО ViPNet Coordinator Linux 4.0 используется новый формат сетевых фильтров и правил трансляции IP-адресов, который является единым для ПО ViPNet (как под управлением ОС Windows, так и ОС Linux), а также позволяет применять политики безопасности, созданные в программе ViPNet Policy Manager 4.0. При обновлении ПО с версии 3.x правила защищенной и открытой сети конвертируются в соответствующие сетевые фильтры и правила трансляции (см. «Обновление версии ПО ViPNet Coordinator Linux» на стр. 33).

- **Использование командного интерпретатора**

  Ранее все настройки правил открытой сети и все параметры фильтрации трафика защищенной сети производились в конфигурационных файлах `firewall.conf` и `iplir.conf` соответственно. Теперь для работы с сетевыми фильтрами и правилами трансляции используется командный интерпретатор (см. «Командный интерпретатор ViPNet Coordinator Linux» на стр. 44).

- **Изменен список поддерживаемых дистрибутивов ОС Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на следующих дистрибутивах ОС Linux:

  - ALT Linux СПТ 7.0;
  - Astra Linux 1.3;
  - CentOS 5.7 (64-разрядная);
  - CentOS 6.4 (32/64-разрядная);
  - RedHat Enterprise Linux 5.7 (64-разрядная);
  - RedHat Enterprise Linux 6.4 AS (32/64-разрядная);
  - SUSE Linux Enterprise Server 10 SP4 (32/64-разрядная);
  - SUSE Linux Enterprise Server 11 SP3 (32/64-разрядная);
  - Ubuntu 12.04.

  Из списка поддерживаемых дистрибутивов ОС Linux были исключены:

  - CentOS 5.4;
  - CentOS 6.0;
  - RedHat Enterprise Linux 5.4;
  - RadHat Enterprise Linux 6.0 AS;
  - SUSE Linux Enterprise Server 10;
  - SLES 11 SP1;
  - Ubuntu 10.04.

- **Изменен список поддерживаемых ядер ОС Linux**

  Теперь минимальная поддерживаемая версия ядра ОС Linux — 2.6.18.

- **Добавлена возможность использования системных групп объектов в сетевых фильтрах и правилах трансляции IP-адресов**

  Ранее все параметры в условии правила необходимо было задавать вручную. Теперь для сетевых фильтров и правил трансляции адресов появилась возможность использовать встроенные системные группы объектов с фиксированными именами (см. «Системные группы объектов» на стр. 86), которые заменяют ряд часто используемых параметров.

- **Отказ от режимов безопасности**

В версии 4.0 режимы безопасности не используются. Необходимый уровень безопасности можно настроить, создав соответствующие сетевые фильтры (см. «Создание сетевого фильтра» на стр. 99).

- **Антиспуфинг**

  Для обеспечения высокого уровня безопасности сети в ViPNet Coordinator Linux используется функция антиспуфинга. В версии 3.2.x настройка антиспуфинга выполнялась администратором вручную. В версии 4.0 настройка антиспуфинга выполняется автоматически (см. «Антиспуфинг» на стр. 107). При этом соответствующие фильтры формируются автоматически на основе таблицы маршрутизации данного сетевого узла.

- **Обработка прикладных протоколов**

  В новой версии ПО ViPNet Coordinator Linux реализована функция обработки следующих прикладных протоколов: FTP, DNS, H.323, SCCP, SIP (см. «Настройка параметров обработки прикладных протоколов» на стр. 119). Данная функция позволяет использовать указанные прикладные протоколы на защищенных узлах, которым назначены виртуальные IP-адреса или для которых выполняется трансляция адресов.

- **Кэширование DNS-имен**

  В ПО ViPNet Coordinator Linux версии 4.0 осуществляется кэширование данных о DNS-именах, полученных от DNS-серверов при добавлении этих имен в фильтры открытой сети. При дальнейшей загрузке данных фильтров в драйвер информация о DNS-именах запрашивается из кэша. Запрос DNS-серверам отправляется, только если данные о DNS-имени не могут быть получены из кэша. Такой подход позволяет ускорить загрузку фильтров в драйвер.

- **Отказ от обновления ключей с помощью высланных из ЦУСа файлов *.dst**

  Автоматическое обновление ключей на ViPNet Coordinator Linux с помощью переданных по сети из программы ViPNet Центр управления сетью дистрибутивов ключей (`*.dst`) больше не поддерживается. Теперь обновление ключей с помощью дистрибутивов ключей возможно только вручную.

Что нового в версии 3.7.4

В этом разделе представлен краткий обзор изменений и новых возможностей ViPNet Coordinator Linux версии 3.7.4.

- **Расширение условий, задаваемых в правилах трансляции IP-адресов**

  Расширен список допустимых выражений для задания в условии правил трансляции IP-адресов. Теперь в лексеме `to` для правил трансляции адреса отправителя и в лексеме `from` для правил трансляции адреса получателя можно указать не только значение `anyip`, но также адрес, диапазон и список адресов, маску адресов, порт и диапазон портов.

- **Возможность указания любого протокола в правилах трансляции IP-адресов**

  Сняты ограничения на протокол, задаваемый в правилах трансляции IP-адресов. Теперь в лексеме `proto` можно указать любой протокол, в том числе протокол GRE. Поддержка

протокола GRE в правилах трансляции позволяет обеспечить доступ удаленных клиентов к серверу по технологии PPTP VPN, которая использует протокол GRE для передачи пакетов.

Что нового в версии 3.7.3

В этом разделе представлен краткий обзор изменений и новых возможностей ViPNet Coordinator Linux версии 3.7.3.

- **Поддержка ядер Linux серии 3.x**

  Ранее в ПО ViPNet Coordinator Linux поддерживались только 32-разрядные ядра Linux серии 2.6.x. Теперь реализована поддержка ядер серии 3.x (от версии 3.0 до версии 3.6 включительно).

- **Запрос пароля при остановке всех демонов и выгрузке всех драйверов ViPNet**

  Ранее аутентификация при остановке всех демонов и выгрузке всех драйверов ViPNet не требовалась. Теперь для обеспечения наибольшей безопасности после ввода команды `vipnet stop` запрашивается пароль пользователя сетевого узла для подтверждения выполнения команды. Команда будет выполнена только в случае ввода правильного пароля пользователя.

- **Добавлена возможность накопления служебного трафика**

  Ранее при возникновении каких-либо событий служебные пакеты сразу же рассылались на узлы. Теперь добавлена функция накопления служебного трафика, обрабатываемого на координаторе, с последующей рассылкой служебных пакетов на узлы не чаще, чем 1 раз в минуту. Использование данной функции позволит уменьшить объем служебного трафика.

- **Возможность гибкой настройки видимости сетевых узлов**

  Изменен принцип настройки видимости сетевых узлов. Раньше видимость узлов (доступ к узлам по реальному или виртуальному IP-адресу) определялась автоматически, и только для отдельных узлов можно было задать принудительный доступ по реальному IP-адресу. Теперь реализована возможность групповой настройки видимости, которая позволяет задать параметры доступа сразу для всех узлов сетей или подсетей ViPNet, а также установить параметры доступа, используемые по умолчанию. Новые настройки видимости задаются в секции `[visibility]` файла `iplir.conf` (см. «Секция [visibility]» на стр. 62).

- **Добавлена возможность выбора метода подсчета контрольной суммы IP-пакета**

  Раньше в ПО ViPNet Coordinator Linux по умолчанию использовался только один метод подсчета контрольной суммы IP-пакета — частично аппаратный подсчет контрольной суммы. В данном методе контрольная сумма служебного заголовка вычисляется программными средствами, а вычисление контрольной суммы остального содержимого пакета осуществляется средствами ядра Linux либо сетевой карты. Теперь появилась возможность использовать метод полностью программного подсчета контрольной суммы. (см. «Выбор метода подсчета контрольной суммы пакета» на стр. 157) В данном методе вычисление контрольной суммы всего IP-пакета (в том числе служебного заголовка) осуществляется программными средствами. Метод подсчета контрольной суммы задается с помощью параметра `hw_csum` в переменной `DRVIPLIR_PARAMS`.

- **Добавлена возможность многопоточной обработки служебных сообщений**

  Раньше в ПО ViPNet Coordinator Linux поддерживался режим многопоточной обработки только сетевого трафика (шифрование и фильтрация) с использованием сразу нескольких процессоров. Теперь появилась поддержка многопоточной обработки служебных сообщений. Для управления потоками обработки служебных сообщений используется параметр `ompnumthreads`, задаваемый в секции `[misc]` файла `iplir.conf`.

Что нового в версии 3.7.1

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.7.1.

- **Поддержка режима шифрования CTR (Counter mode)**

  Реализован режим шифрования CTR (в дополнение к используемому режиму CFB), который позволяет получить выигрыш в скорости до 20%. Теперь можно выбрать нужный режим с помощью отдельного параметра, передаваемого криптографическому драйверу при его загрузке. По умолчанию используется режим CFB (Cipher Feedback).

- **Создана версия справочного руководства в виде man-страниц**

  В предыдущих версиях ПО в драйвер ViPNet не были включены man-страницы. Теперь для удобства просмотра справочное руководство можно вызвать прямо из командной консоли.

Что нового в версии 3.7.0

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.7.0.

- **Оптимизирован список поддерживаемых дистрибутивов ОС Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на следующих дистрибутивах ОС Linux:

  - ALT Linux 6.0 Server;
  - ALT Linux 6.0 Desktop;
  - CentOS 5.4;
  - CentOS 5.7;
  - CentOS 6.0;
  - Mandriva Linux 2010 Powerpack;
  - RHEL 5.7;
  - RHEL 6.0 AS;
  - SLES 11 SP1;
  - Ubuntu 10.04.

Из списка поддерживаемых дистрибутивов ОС Linux исключены следующие устаревшие (не используемые на практике) дистрибутивы ОС Linux:

- ALT Linux 4.0 Server;
- ALT Linux 4.0 Desktop;
- Debian Lenny 5.0;
- OpenSUSE 11.2 Desktop;
- RedHat Enterprise Linux 4.0 AS;
- Slackware Linux 10.2;
- Ubuntu 8.04 LTS Desktop;
- Ubuntu 9.10 Desktop.

- **Изменен список поддерживаемых ядер ОС Linux**

  Ядра версий 2.4.x более не поддерживаются, поддержка ядер версий 2.6.x расширена до версии 2.6.39 включительно.

- **Изменена логика использования 5-го режима безопасности**

  Исключена возможность установки 5-го режима безопасности на отдельных сетевых интерфейсах. Теперь этот режим можно установить только одновременно на всех сетевых интерфейсах (см. «Настройка параметров сетевых интерфейсов» на стр. 66). Включение и отключение 5-го режима безопасности на интерфейсах производится с помощью специальных команд, приведенных в главе Управление ViPNet Coordinator Linux (на стр. 43).

- **Поддержка многопоточности по умолчанию**

  В предыдущих версиях ПО в драйвере ViPNet по умолчанию устанавливался однопоточный режим обработки трафика. Теперь по умолчанию устанавливается многопоточный режим с числом потоков, равным числу процессоров. Использование многопоточного режима позволяет существенно увеличить скорость обработки трафика, однако пользователь по-прежнему может изменить число используемых потоков (см. «Управление числом потоков в драйвере» на стр. 155).

Что нового в версии 3.6.4

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.6.4.

- **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. 61). Теперь генерируемые виртуальные адреса не могут превышать значения, заданного этим параметром.

Что нового в версии 3.6.3

В версии 3.6.3 исправлен ряд ошибок, выявленных в процессе эксплуатации версии 3.6.2.

Что нового в версии 3.6.2

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.6.2.

- **Поддержка ограничения срока действия лицензии на ПО ViPNet Coordinator Linux фиксированной датой**

  Реализована проверка лицензионных ограничений на дату окончания действия лицензии. Проверка осуществляется всеми службами ПО ViPNet Coordinator Linux (iplircfg, mftpd, failoverd) при старте и раз в сутки. При обнаружении истечения срока действия лицензии служба останавливается и выводит соответствующее сообщение на консоль (если это старт службы) и в системный журнал ОС Linux. Если истечение срока действия лицензии обнаружено демоном iplircfg, то он не только останавливается сам, но и останавливает другие службы (mftpd, failoverd), а также выгружает драйверы ViPNet.

Что нового в версии 3.6.1

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.6.1.

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

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

Что нового в версии 3.6.0

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.6.0.

- **Оптимизирован список поддерживаемых дистрибутивов ОС Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на следующих дистрибутивах ОС Linux:

  - RedHat Enterprise Linux 5.4;
  - OpenSUSE 11.2 Desktop;

- SuSe Linux Enterprise Server 10 SP3;
- Debian Lenny 5.0;
- Ubuntu 9.10 Desktop.

Из списка поддерживаемых дистрибутивов ОС Linux исключены следующие устаревшие (не используемые на практике) дистрибутивы ОС Linux:

- SuSe Linux 10.0;
- Debian Etch 4.0 r1;
- OpenSuSe 11.1;
- Linux XP Server 2008;
- Linux XP Desktop 2008 Secure Edition.

- **Расширен список поддерживаемых ядер Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на версиях ядра до 2.6.32 включительно.

- **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 координаторы, работающие под управлением различных ОС, имеют одинаковую логику поведения.

- **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.

- **Интеграция с Системой оценки защищенности (СОЗ) ЦБИ**

  Реализован автоматический экспорт информации об узле с ПО ViPNet Coordinator Linux для СОЗ ЦБИ. Экспорт выполняется отдельно поставляемой утилитой, которую можно запускать на регулярной основе.

- **Поддержка многопоточности в драйвере ViPNet**

  Реализована поддержка многопоточности в драйвере ViPNet. Теперь можно управлять числом потоков с помощью отдельного параметра, передаваемого драйверу при его загрузке.

Для совместимости с предыдущими версиями по умолчанию установлен однопоточный режим.

Что нового в версии 3.5.2

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.5.2.

- **Расширенные возможности настройки с помощью апплета SGA**

  Реализована поддержка настройки фильтров открытой сети и режимов безопасности с помощью апплета мониторинга и управления SGA, а также поддержка авторизации доступа с помощью SGA.

- **Защита пароля ViPNet**

  Реализовано хранение пароля ViPNet в защищенном виде.

- **Сняты ограничения на управление и мониторинг с помощью апплета SGA**

  Удалена проверка прикладной задачи «Сервер SGA» в ПО ViPNet Coordinator Linux. Теперь для управления и мониторинга с помощью SGA узел с ПО ViPNet Coordinator Linux не надо регистрировать в прикладной задаче «Сервер SGA».

Что нового в версии 3.5.1

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.5.1.

- **Корректная аутентификация транспортного модуля MFTP при взаимодействии с другими версиями MFTP**

  Доработан транспортный модуль MFTP в составе ViPNet Coordinator Linux для корректной аутентификации с MFTP, работающим в составе других приложений ViPNet. Как следствие доработки, упразднен параметр `auth_type` в секции `[misc]` файла конфигурации `mftp.conf`.

- **Восстановлена совместимость со старыми версиями ПО ViPNet**

  Возвращена поддержка старого формата 4.0 для исходящих пакетов. Теперь узел с ПО ViPNet Coordinator Linux может взаимодействовать с узлами, на которых установлены старые версии ПО ViPNet (без поддержки нового формата 4.1).

Что нового в версии 3.5.0

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.5.0.

- **Расширен список поддерживаемых дистрибутивов ОС Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на следующих дистрибутивах ОС Linux:

- OpenSUSE 11.1;
- Linux XP Desktop 2008 Secure Edition;
- Linux XP Server 2008.

- **Расширен список поддерживаемых ядер Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на версиях ядра до 2.6.29 включительно.

- **Поддержка централизованной политики безопасности**

  Реализована поддержка централизованной политики безопасности. Теперь узел с ПО ViPNet Coordinator Linux может принимать и обрабатывать политики безопасности, присланные из Центра управления политиками безопасности (программы ViPNet PolicyManager).

Что нового в версии 3.4.1

В версии 3.4.1 исправлен ряд ошибок, выявленных в процессе эксплуатации версии 3.4.0.

Что нового в версии 3.4.0

В этом разделе представлен краткий обзор изменений и новых возможностей версии 3.4.0.

- **Расширен список поддерживаемых дистрибутивов ОС Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на следующих дистрибутивах ОС Linux:

  - SuSe Linux Enterprise Server 10 SP1, SP2;
  - Ubuntu 8.04 LTS Desktop;
  - Debian Etch 4.0 r1.

- **Расширен список поддерживаемых ядер Linux**

  Реализована поддержка функционирования ПО ViPNet Coordinator Linux на версиях ядра до 2.6.24 включительно.

- **Расширен список поддерживаемых алгоритмов шифрования**

  Реализована поддержка алгоритма шифрования AES. Теперь администратор может выбрать используемый алгоритм для исходящих сетевых пакетов — ГОСТ или AES.

- **Поддержка многоядерности сетевым драйвером ViPNet**

  Реализована поддержка сетевым драйвером ViPNet многоядерных процессоров Intel, что позволяет выполнять параллельную обработку шифрованных пакетов.

- **Более простая настройка передачи файлов в кластере горячего резервирования**

Изменена настройка резервирования служебных файлов ViPNet при работе в режиме кластера. Теперь не надо перечислять резервируемые файлы: файлы разбиты на логические группы, и за включение и отключение резервирования каждой группы отвечает отдельный параметр.

- **Оптимизирован механизм защиты от сбоев в кластере горячего резервирования**

  Реализована дополнительная защита от сбоев на уровне сетевого драйвера ViPNet.

- **Расширенные возможности поиска в журнале регистрации IP-пакетов**

  Реализован поиск узла отправителя и получателя при задании фильтра в консольной версии программы просмотра журнала регистрации IP-пакетов.

- **Поддержка системы централизованного мониторинга ViPNet StateWatcher**

  Реализованы команды для передачи информации о состоянии узла с ПО ViPNet Coordinator Linux в систему централизованного мониторинга.

- **Интеграция в состав сети ViPNet OFFICE**

  Появилась возможность использовать узел с ПО ViPNet Coordinator Linux в качестве координатора защищенной сети, созданной на базе пакета ViPNet OFFICE.

J

Глоссарий

ViPNet Administrator

Набор программного обеспечения для администрирования сети ViPNet, включающий в себя серверное и клиентское приложения ViPNet Центр управления сетью, а также программу ViPNet Удостоверяющий и ключевой центр.

ViPNet Policy Manager

Программа, которая входит в состав программного комплекса ViPNet. Предназначена для централизованного управления политиками безопасности узлов защищенной сети ViPNet.

Антиспуфинг

Защита от спуфинг-атак, при которых злоумышленник подделывает адрес источника для обхода межсетевых экранов и организации DoS-атак (от англ. Denial of Service, отказ в обслуживании).

Виртуальная защищенная сеть

Технология, позволяющая создать логическую сеть, чтобы обеспечить множественные сетевые соединения между компьютерами или локальными сетями через существующую физическую сеть. Уровень доверия к такой виртуальной сети не зависит от уровня доверия к физическим сетям благодаря использованию средств криптографии (шифрования, аутентификации и средств персонального и межсетевого экранирования).

Виртуальный IP-адрес

IP-адрес, который приложения на сетевом узле ViPNet (А) используют для обращения к ресурсам сетевого узла ViPNet (Б) или туннелируемых им узлов вместо реального IP-адреса узла. Виртуальные IP-адреса узлу ViPNet Б назначаются непосредственно на узле А. На других узлах узлу

ViPNet Б могут быть назначены другие виртуальные адреса. Узлу ViPNet (Б) назначается столько виртуальных адресов, сколько реальных адресов имеет данный узел. При изменении реальных адресов у узла Б выделенные ему виртуальные адреса не изменяются. Виртуальные адреса туннелируемых узлов привязываются к реальным адресам этих узлов и существуют, пока существует данный реальный адрес. Использование виртуальных адресов позволяет избежать конфликта реальных IP-адресов в случае, если узлы работают в локальных сетях с пересекающимся адресным пространством, а также использовать эти адреса для аутентификации удаленных узлов в приложениях ViPNet.

Дистрибутив ключей

Файл с расширением `.dst`, создаваемый в программе ViPNet Удостоверяющий и ключевой центр для каждого пользователя сетевого узла ViPNet. Содержит справочники, ключи и файл лицензии, необходимые для обеспечения первичного запуска и последующей работы программы ViPNet на сетевом узле. Для обеспечения работы программы ViPNet дистрибутив ключей необходимо установить на сетевой узел.

Защищенный узел

Сетевой узел, на котором установлено программное обеспечение ViPNet с функцией шифрования трафика на сетевом уровне.

Открытый Интернет

Технология, реализованная в программном обеспечении ViPNet. При подключении к Интернету узлы локальной сети изолируются от сети ViPNet, а при работе в сети ViPNet — от Интернета, что обеспечивает защиту от возможных сетевых атак извне без физического отключения компьютеров от локальной сети.

Политика безопасности

Набор параметров, регулирующих безопасность сетевого узла. В технологии ViPNet безопасность сетевых узлов обеспечивается с помощью сетевых фильтров и правил трансляции IP-адресов.

Справочники и ключи

Справочники, ключи узла и ключи пользователя.

Трансляция сетевых адресов (NAT)

Технология, позволяющая преобразовывать IP-адреса и порты, использующиеся в одной сети, в адреса и порты, использующиеся в другой.

Транспортный конверт

Зашифрованная информация служб или приложений, доставляемая на сетевые узлы ViPNet транспортным модулем ViPNet MFTP.

Туннелирование

Технология, позволяющая защитить соединение с участием открытых узлов при передаче данных через Интернет и другие публичные сети. Туннелирование заключается в шифровании трафика открытых узлов координаторами.

Туннелируемый узел

Узел, на котором не установлено программное обеспечение ViPNet с функцией шифрования трафика на сетевом уровне, но его трафик на потенциально опасном участке сети зашифровывается и расшифровывается на координаторе, за которым он стоит.

Шлюзовой координатор

Координатор, через который осуществляется обмен транспортными конвертами между сетями ViPNet, установившими межсетевое взаимодействие.

Шлюзовые координаторы назначаются в ЦУСе каждой сети при организации взаимодействия между двумя различными сетями ViPNet.

Указатель

V

ViPNet Policy Manager - 18

А

Антиспуфинг - 79, 195, 201

В

Виртуальная защищенная сеть - 18
Виртуальный IP-адрес - 62
Выбор метода подсчета контрольной суммы пакета - 202

Ж

Журналы устранения неполадок ПО ViPNet Coordinator Linux - 13, 65, 131

И

Изменение сетевого фильтра - 100, 118
Изменения, производимые в системе при установке и удалении ViPNet Coordinator Linux - 32
Использование групп объектов - 198
Использование патча ядра - 13, 22, 23, 69
Использование технологии NETFILTER - 23, 69
История версий - 15

К

Командный интерпретатор ViPNet Coordinator Linux - 82, 200
Команды группы alg - 44
Команды группы firewall - 44
Команды группы iplir option - 44
Компоненты сетевых фильтров - 82, 99, 115
Конвертация правил защищенной сети при обновлении до версии 4.x - 33
Конвертация правил открытой сети при обновлении до версии 4.x - 33

М

Методы перехвата сетевых пакетов - 22

Н

Настройка параметров защищенного подключения - 11, 31
Настройка параметров защищенной сети - 167
Настройка параметров обработки прикладных протоколов - 11, 169, 196, 201
Настройка параметров сетевых интерфейсов - 57, 204
Настройка параметров фильтрации IP-трафика - 34
Настройка режима - 52, 53, 56, 57, 58, 63
Настройка режимов работы через межсетевой экран - 50, 52, 55, 57
Настройка системы защиты от сбоев - 128

О

Обновление версии ПО ViPNet Coordinator Linux - 81, 199
Общие принципы настройки - 30
Общие сведения о сетевых фильтрах - 79, 120

П

Переход от использования фиксированных альтернативных каналов к множественным адресам доступа - 199
Пользовательские группы объектов, настроенные по умолчанию - 85, 96, 197
Правила определения видимости сетевых узлов - 53, 62
Примеры настройки туннелей с использованием координаторов ViPNet - 54
Принципы назначения виртуальных адресов - 49, 53, 61, 159
Программа просмотра журнала регистрации IP-пакетов - 43, 53
Программа просмотра информации о защищенном узле - 30, 43
Программа просмотра информации о состоянии системы защиты от сбоев - 130
Программа работы с конфигурациями ViPNet - 43, 60
Программа распаковки дистрибутива ключей - 46
Программа смены пароля пользователя - 43
Просмотр сетевых фильтров - 100, 106, 118
Просмотр, изменение, удаление правил трансляции адресов - 117
Процессы-демоны, входящие в состав ПО ViPNet Linux - 31

Р

Расписание - 90, 198
Ручное конфигурирование ViPNet Coordinator Linux - 22, 28, 29
Ручное переназначение виртуальных адресов узлов - 62

С

Сбор информации о состоянии ПО ViPNet с использованием протокола SNMP - 11
Секция [id] - 64, 197, 198
Секция [misc] - 186, 197
Секция [virtualip] - 64, 159, 204
Секция [visibility] - 53, 64, 202
Система защиты от сбоев - 11, 31, 142, 169
Системные группы объектов - 34, 82, 85, 95, 115, 200
Системные требования - 22, 27, 161
События, отслеживаемые ПО ViPNet Coordinator Linux - 26
Создание правила трансляции IP-адресов - 90, 112
Создание сетевого фильтра - 34, 37, 90, 116, 201
Состав системы защиты от сбоев и принципы ее работы - 11, 131

Т

Трансляция адреса источника - 111
Трансляция адреса назначения - 111
Трансляция сетевых адресов (NAT) - 20, 68, 112
Транспортный конверт - 19

У

Удаление сетевого фильтра - 100, 118
Управление ViPNet Coordinator Linux - 204
Управление числом потоков в драйвере - 172, 204
Условие - 90
Установка ПО ViPNet Coordinator Linux - 22, 33

Ф

Фрагментирование шифрованных ViPNet-пакетов - 60
Функции координатора в защищенной сети ViPNet - 11