Межсетевой экран «Z-2» 2.4
Руководство администратора

Инфосистемы Джет
Москва
2006

Оглавление

Список иллюстраций

Список таблиц

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ С КОНЕЧНЫМ ПОЛЬЗОВАТЕЛЕМ

Настоящее Лицензионное Соглашение является юридическим соглашением, заключаемым между конечным пользователем (юридическим или физическим лицом), и ЗАО «Инфосистемы Джет» (далее *компания*), обладающее авторскими правами на ПРОДУКТ, который поставляется вместе с этим Лицензионным Соглашением. ПРОДУКТ включает в себя записанную на соответствующих носителях или полученную через Интернет программу для ЭВМ, любую связанную с ПРОДУКТОМ «встроенную», «электронную», получаемую через Интернет или печатную документацию. Используя данный ПРОДУКТ (в том числе устанавливая, копируя, загружая, осуществляя доступ или используя его иным образом), конечный пользователь тем самым соглашается с условиями настоящего лицензионного соглашения. Если конечный пользователь не согласен с условиями, то он не имеет права использовать данный ПРОДУКТ.

Основные понятия

Основные понятия, используемые в настоящем Лицензионном Соглашении:

ПРОДУКТ программа для ЭВМ Межсетевой экран Z-2; прилагаемые к ним печатная и любая другая «встроенная», «электронная» или получаемая через Интернет документация.

ПО представленная в объектном коде программа для ЭВМ.

Документация печатная и любая другая «встроенная», «электронная» или получаемая через Интернет документация, прилагаемая к ПО.

Использовать ПРОДУКТ копировать, устанавливать, осуществлять доступ, запускать, отображать или иным образом осуществлять взаимодействие с ПРОДУКТОМ.

Конечный пользователь юридическое или физическое лицо, которому предоставляется ограниченное данным Лицензионным Соглашением право использовать ПРОДУКТ.

Обозначенный компьютер любая ЭВМ, компьютер, сервер, рабочая станция или другое устройство, на которые устанавливается ПРОДУКТ и на котором он используется.

Лицензионное Соглашение или Лицензия или Соглашение Соглашение между конечным пользователем и компанией, которое предоставляет конечному пользователю ограниченное право использовать ПРОДУКТ.

Адрес зарегистрированный IP-адрес в сети конечного пользователя.

Адаптация ПО внесение изменений, осуществляемых исключительно в целях обеспечения функционирования ПО на конкретных технических средствах конечного пользователя или под управлением конкретных программ конечного пользователя.

Модификация (переработка) ПО любые изменения ПО, не являющиеся адаптацией.

Декомпилирование ПО технический прием, включающий преобразование объектного кода в исходный текст в целях изучения структуры и кодирования ПО.

Воспроизведение ПО изготовление одного или более экземпляров ПО в любой материальной форме, а также их запись в память ЭВМ.

Распространение ПО предоставление третье стороне доступа к воспроизведенному в любой материальной форме ПО, в том числе сетевыми и иными способами, а также путем продажи, проката, сдачи внаем, предоставления взаймы, включая импорт для любой из этих целей.

Выпуск в свет (опубликование) ПРОДУКТА предоставление экземпляров ПРОДУКТА неопределенному кругу лиц (в том числе путем записи в память ЭВМ и выпуска печатного текста).

Объем Лицензии

Компания, как обладатель авторских прав, предоставляет конечному пользователю Лицензию на данный ПРОДУКТ, т.е. неэксклюзивное право на его использование только тем способом и на тех условиях, которые указаны в настоящем Лицензионном Соглашении. Все условия, оговоренные далее, относятся как к ПРОДУКТУ в целом, так и ко всем его компонентам в отдельности.

Установка и использование

Данное соглашение дает конечному пользователю право устанавливать и использовать одну копию ПО, входящего в ПРОДУКТ, и относящейся к нему Документации. Копия ПО может быть установлена только на один обозначенный компьютер. Данное ограничение не распространяется на ПО Третьих производителей, установка которых может осуществляться на другой компьютер.

Плата за Лицензию

Чтобы иметь право использовать ПРОДУКТ конечный пользователь должен оплатить полную стоимость Лицензии. Данный ПРОДУКТ лицензируется по количеству АДРЕСОВ защищаемой сети конечного пользователя. Если количество АДРЕСОВ сети конечного пользователя увеличивается и становится больше количества оплаченных лицензий, то конечный пользователь обязан приобрести у компании дополнительные лицензии на ПРОДУКТ.

Использование ПО, поставляемого Третьей стороной

Для повышения функциональных возможностей конечный пользователь может использовать совместно с ПРОДУКТОМ ПО, поставляемое ему Третьей стороной. Использование данного ПО регламентируется лицензионными соглашениями, заключенными между конечным пользователем и представителями Третьей стороны. Установка данного ПО может быть осуществлена как на обозначенный компьютер, так и на другой компьютер.

Защита интеллектуальной собственности

В соответствии с Федеральным Законом от 9 июля 1993 г. 85-ФЗ «Об авторском праве и смежных правах», Законом РФ от 23 сентября 1992 г. 3523-1 «О правовой охране программ для электронных вычислительных машин и баз данных» ПРОДУКТ является объектом правовой охраны. Конечный пользователь принимает и соглашается с тем, что компания обладает исключительными имущественными правами на ПРОДУКТ и связанную с ним Документацию, включая любые их модификации, усовершенствованные версии и копии. Лицензионное Соглашение не передает конечному пользователю имущественные права на ПРОДУКТ, а только предоставляет ему ограниченное право использования, которое может быть отменено в соответствии с условиями данного Лицензионного соглашения. За нарушение авторских прав на ПРОДУКТ нарушитель несет гражданскую, административную и/или уголовную ответственность в соответствии с законодательством РФ.

Ограничения прав конечного пользователя

1. Конечный пользователь вправе осуществлять адаптацию ПО исключительно в целях обеспечения функционирования ПРОДУКТА.
2. Устанавливая ПРОДУКТ на свой компьютер, конечный пользователь может сделать одну архивную копию ПРОДУКТА, применяемую для замены правомерно приобретенного экземпляра ПРОДУКТА в случае, когда его оригинал утерян, уничтожен или стал непригодным для использования. При этом конечный пользователь должен соблюдать условие, что такая копия не может быть использована для иных целей и подлежит уничтожению в случае, когда дальнейшее использование ПРОДУКТА перестает быть правомерным. Кроме того, архивная копия ПРОДУКТА должна включать упоминание об авторских и имущественных правах компании, включая копию настоящего Лицензионного соглашения.
3. Конечному пользователю запрещено без соответствующего соглашения с компанией:
   - выпускать в свет ПРОДУКТ;
   - воспроизводить ПРОДУКТ (полностью или частично) в любой форме, любыми способами;
   - модифицировать ПО, в том числе переводить ПО с одного языка на другой;
   - распространять ПРОДУКТ иными способами, чем предусмотрено настоящим Лицензионным Соглашением;
   - использовать ПРОДУКТ не для нужд, указанных в настоящем Лицензионном Соглашении.

Передача ПРОДУКТА третьим лицам

Конечный пользователь может передать все свои права по данному Соглашению другому физическому или юридическому лицу при условии, что первоначальный пользователь передает все компоненты ПРОДУКТА без исключения, а получатель согласен с условиями настоящего Соглашения. Сразу после передачи первоначальный пользователь обязан прекратить использование ПРОДУКТА и удалить все его компоненты со всех компьютеров либо возвратить компании.

Сроки Соглашения и прекращение Лицензии

Срок действия Соглашения начинается с момента начала установки ПРОДУКТА на обозначенный компьютер конечного пользователя и продолжается без ограничения времени или до тех пор, пока одна из сторон не расторгнет настоящее Соглашение. Лицензия может быть прекращена при нарушении одной из сторон условий настоящего Соглашения. Компания может прекратить действие данной Лицензии в случае, если конечный пользователь нарушит хотя бы один из пунктов настоящего Соглашения. Прекращение Лицензии конечного пользователя предусматривает обязательное и немедленное прекращение использования ПРОДУКТА, удаление всех его компонентов со всех компьютеров конечного пользователя либо возвращение его компании.

Ограничение гарантий

1. ПРОДУКТ продается без гарантий относительно его коммерческой ценности. Компания не гарантирует соответствия характеристик ПРОДУКТА условия или целям его использования конечным пользователем.
2. Компания не гарантирует того, что ПРОДУКТ не содержит ошибок. Наличие таких ошибок не должно составлять причину нарушения конечным пользователем данного Соглашения. При этом компания обязуется рассмотреть причины, вызвавшие сбой или неполадки и, при возможности, учесть их в новой версии ПРОДУКТА.

3. Если носитель, поставляемый компанией, вышел из строя (не по вине конечного пользователя и при условии надлежащей эксплуатации носителя в соответствии с функциональными требованиями, описанными в Документации), конечный пользователь обязан проинформировать компанию об этом, после чего компания осуществляет бесплатную замену вышедшего из строя носителя.
4. За исключением ограниченных гарантий, описанных в этом разделе, не существует каких-либо других явно выраженных или подразумеваемых гарантий, предоставляемых конечному пользователю в соответствии с данным Соглашением.

Другие исключаемые услуги

Плата за Лицензию не включает расходы на следующие виды услуг:

а. установку;
б. наладку;
в. конфигурацию ПО и общую настройку системы;
г. техническую поддержку и устранение неполадок;
д. обучение;
е. настройку и улучшение работы системы;
ж. усовершенствование работы ПО.

Проверка выполнения Соглашения

Конечный пользователь обязан осуществлять контроль над использованием ПРОДУКТА в соответствии с условиями Лицензионного соглашения и обеспечить безопасное хранение ПРОДУКТА. Настоящим Соглашением конечный пользователь предоставляет право компании в соответствии с действующим законодательством осуществлять проверку выполнения конечным пользователем взятых на себя в соответствии с настоящим Лицензионным Соглашением обязательств.

Ответственность конечного пользователя за нарушение условий и ограничений настоящего Лицензионного Соглашения

В случае нарушения конечным пользователем условий и ограничений настоящего Лицензионного Соглашения он является нарушителем авторского права. Нарушением исключительного права на ПРОДУКТ является не санкционированное компанией использование ПРОДУКТА путем выпуска ПО в свет; воспроизведения (полного или частичного) в любой форме, любыми способами; распространение; модификация, в том числе перевод с одного языка на другой; иное использование ПРОДУКТА. В случае нарушения авторских прав на ПРОДУКТ компания оставляет за собой право обратиться с иском в арбитражный и/или иной суд в соответствии с действующим законодательством РФ.

Полнота Лицензионного Соглашения

Данное Лицензионное Соглашение представляет собой полное Соглашение между сторонами и отменяет все предыдущие или одновременные соглашения или договоренности, устные или письменные. Все вопросы, связанные с использованием ПРОДУКТА и данным Лицензионным Соглашением, следует направлять по адресу:

ЗАО "Инфосистемы Джет"
Россия,127015, Москва
ул. Большая Новодмитровская, д. 14/1
тел. (095) 411 76 01
факс (095) 411 76 02
e-mail: z-2@jet.msk.su

Предисловие

Для кого предназначен этот документ

Данное руководство предназначено для администраторов сетей или администраторов безопасности, ответственных за обеспечение политики безопасности организации в части разграничения доступа к компьютерной сети.

Руководство подразумевает наличие знаний и рабочих навыков в следующем объеме:

- знания стека протоколов TCP/IP;
- базовые навыки администрирования Solaris.

Краткое содержание

- *«Введение»* предоставляет краткие сведения о МЭ «Z-2», о составе компонент и вариантах поставки.
- *«Инсталляция межсетевого экрана «Z-2»»* описывает процедуру подготовки ОС к инсталляции межсетевого экрана, инсталляцию и порядок запуска МЭ.
- *«Архитектура межсетевого экрана «Z-2»»* дает представление об архитектуре МЭ и вариантах фильтрации информационных потоков с помощью МЭ.
- *«Пакетный фильтр»* описывает компоненту МЭ «пакетный фильтр» — принципы его работы, варианты фильтрации, систему трансляции адресов, а также утилиты командной строки для управления пакетным фильтром.
- *«Прикладные шлюзы»* описывает прикладные шлюзы, входящие в состав МЭ, — их архитектуру, принципы их работы и особенности каждого прикладного шлюза.
- *«Сервер аутентификации»* предоставляет сведения о сервере аутентификации и принципах работы и использования подключаемых модулях аутентификации (PAM).
- *«Протоколирование информации»* описывает формат протоколов от различных подсистем межсетевого экрана.
- *«Анализ регистрационной информации»* описывает утилиты для анализа регистрационной информации, полученной от пакетного фильтра и прикладных шлюзов МЭ.
- *«Подсистема контроля целостности»* представляет сведения о принципах функционирования и способах настройки подсистемы контроля целостности информационной части МЭ.
- *«Утилиты для сохранения и восстановления конфигурации МЭ»* описывает утилиты для оперативного сохранения и восстановления конфигурации МЭ.
- *«Проверка на вирусы с использованием Symantec ScanEngine»* представляет сведения о способах интеграции МЭ и систем антивирусной защиты.

Глава 1

Введение

Межсетевой экран «Z-2» (далее МЭ «Z-2») представляет собой программное средство сетевого разграничения доступа, контролирующее информационные потоки между сетями или фрагментами IP-сетей.

МЭ реализует контроль поступающей и выходящей информации и обеспечивает защиту сети посредством фильтрации данных, т.е. их анализа по совокупности критериев и принятия решения о дальнейшем распространении информации в (из) информационной системы.

МЭ «Z-2» состоит из следующих составных частей:

- Пакетный фильтр
- Набор прикладных шлюзов
- Сервер аутентификации и авторизации
- Графический интерфейс администратора
- Управляющий сервер
- Подсистемы контроля целостности
- Подсистемы анализа регистрационной информации

МЭ поставляется в следующей комплектации:

- Графический интерфейс администратора
- Управляющий сервер
- Пакетный фильтр
- Сервер аутентификации и авторизации
- Подсистема контроля целостности
- Прикладной шлюз протокола HTTP
- Прикладной шлюз протокола SMTP
- Прикладной шлюз протокола POP3
- Прикладной шлюз протокола FTP
- Прикладной шлюз протокола TCP
- Прикладной шлюз протокола UDP
- Прикладной шлюз протокола TELNET
- Прикладной шлюз протокола SNMP
- Прикладной шлюз протокола Net8

Дополнительные сведения о МЭ «Z-2» могут быть получены с сайта компании «Инфосистемы-Джет»: http://www.jetsoft.ru/product/z-2/purpose.html.

Адрес электронной почты службы поддержки продукта: z-2-support@jet.msk.su.

Перед обращением в службу поддержки настоятельно рекомендуется с помощью утилиты `report.sh` собрать информацию об установленных в системе пакетах. Это позволит более точно

диагностировать проблему. Подробная информация об утилите `report.sh` приведена в разделе 12.3 на стр. 163).

Глава 2

Инсталляция межсетевого экрана «Z-2»

2.1 Общие положения

ПО межсетевого экрана выпускается для следующих операционных систем:

- Solaris 9 SPARC
- Solaris 9 x86

Процедура установки межсетевого экрана состоит из следующих шагов:

- подготовка ОС `Solaris` к установке МЭ «Z-2»;
- установка межсетевого экрана «Z-2»;
- установка графического интерфейса для управления МЭ «Z-2».

Процедура установки межсетевого экрана «Z-2» описана в разделе 2.5 на стр. 29 данного руководства. Описание процедуры инсталляции графического интерфейса описана в руководстве по графическому интерфейсу (*«Универсальный графический интерфейс 2.4 Руководство администратора»*, раздел *«Инсталляция графического интерфейса»*).

2.2 Требования к системе

Установка ПО МЭ «Z-2» осуществляется на выделенный компьютер, специально выделенный в качестве межсетевого экрана. Использование выделенного компьютера для решения задач, не связанных с функцией разграничения сетевого доступа, не допускается. В частности, по соображениям безопасности, не допускается установка на выделенный компьютер дополнительного ПО кроме описанного в настоящем руководстве(например, ПО для разработки программ) или разрешение на выделенном компьютере дополнительных сетевых сервисов кроме тех, что предоставляются межсетевым экраном и определяются политикой безопасности компании-владельца межсетевого экрана.

Компания «Инфосистемы Джет» настоятельно рекомендует производить установку графического интерфейса управления МЭ «Z-2» на отдельный компьютер (рабочее место администратора межсетевого экрана). Это позволит повысить безопасность межсетевого экрана и отказаться от использования потенциально небезопасной X-Windows-System на компьютере, на котором установлен межсетевой экран.

Для успешного и бесперебойного функционирования ПО МЭ «Z-2» компьютер, на который осуществляется установка ПО, должен соответствовать аппаратным требованиям для ОС Solaris. Список поддерживаемого оборудования для платформы Intel i386 можно получить через Интернет, обратившись на страницу www.sun.com/bigadmin/hcl/.

Система, на которую планируется установить МЭ «Z-2», должна удовлетворять следующим требованиям:

- Не менее 256 мегабайт оперативной памяти
- 250 мегабайт свободного дискового пространства в директории */opt*
- 3 мегабайта свободного дискового пространства в директории */usr* и */sbin*

2.3 Подготовка к инсталляции

2.3.1 Подготовка ОС `Solaris`

Перед установкой МЭ на ОС `Solaris` следует установить собственно операционную систему Solaris 9, а также рекомендуется установить последние программные патчи и обновления.

Порядок установки ОС Solaris изложен в руководстве по инсталляции операционной системы Solaris.

2.3.1.1 Установка программных обновлений ОС `Solaris`

Установка программных обновлений производится с дисков `SunSolve`, доступных по подписке для зарегистрированных пользователей компании `SUN Microsystems`. Вместе с носителями в комплект поставки `SunSolve` входит руководство по установке программных дополнений. За дополнительной информацией по установке программных дополнений следует обратиться к этому руководству. Также программные обновления для ОС Solaris доступны на сайте http://sunsolve.sun.com.

2.3.2 Подготовка к инсталляции ПО МЭ «Z-2».

Перед началом инсталляции ПО МЭ «Z-2» следует еще раз убедиться в том, что требования, изложенные выше в разделе 2.2 на стр. 25, выполнены и проделана вся подготовительная работа в части подготовки ОС Solaris.

Кроме этого, следует убедиться в отсутствии уже установленной предыдущей версии ПО МЭ «Z-2» или предыдущей инсталляции устанавливаемой версии. Если на диске обнаружена предыдущая версия ПО МЭ «Z-2», то следует произвести ее деинсталляцию. Наличие установленной версии ПО МЭ «Z-2» можно проверить выполнив команду:

```
/usr/bin/pkginfo | grep JET
```

Если список инсталляционных пакетов, найденный указанной последовательностью команд не пуст и среди них есть пакет *JETfw*, то ПО МЭ «Z-2» было установлено полностью или частично. Процедура деинсталляции описана в разделе 2.5.4 на стр. 32.

Перед установкой ПО МЭ «Z-2» следует определить IP-адрес компьютера, на который будет установлен или уже установлен графический интерфейс (IP-адрес рабочего места администратора межсетевого экрана), с которого будет осуществляться управление МЭ «Z-2». Знание адреса потребуется при установке МЭ «Z-2».

2.4 Состав дистрибутивного носителя

ПО межсетевого экрана «Z-2» поставляется на носителе типа CDROM. Для каждой из поддерживаемых ОС поставляется отдельный носитель. CDROM имеет стандартный формат ISO 96660 (high sierra file system), который является стандартным как для поддерживаемых операционных систем, так и для ОС Windows, Linux многих других.

Дистрибутивный носитель имеет следующую структуру каталогов:

```
cdrom
|
|
+--docs/ - Каталог, содержащий файлы с документацией.
|  + Z2AdminGuide.pdf
|  + GUIAdminGuide.pdf
|
+--gui/ - Каталог, содержащий инсталляционный пакет графического
|  +     интерфейса.
|  + BUILD
|  + demomserv
|  + demomserv/JETZ2Demo-Linux-<версия>.sh
|  + demomserv/JETZ2Demo-Windows-<версия>.exe
|  + JETGUI-generic_unix-<версия>.sh
|  + JETGUI-linux-<версия>.sh
|  + JETGUI-solaris_sparc-<версия>.sh
|  + JETGUI-windows-<версия>.exe
|  + JETGUI-Mac_OS_X_Single_Bundle-<версия>.dmg
|
+--README
|
+--sources/ - Каталог, содержащий исходные тексты.
|  + COPYING
|  + z2core.tar.gz
|
+--Z-2-i386-pc-solaris2.9/ - Каталог, содержащий инталляционные пакеты
|  +                         МЭ "Z-2" для Solaris 9 x86
|  + BUILD
|  + fwinstall.sh
|  + fwunattended.sh
|  + fwuninstall
|  + iengine.sh
|  + ipf.pkg
|  + JETcfgen.pkg
|  + JETfile.pkg
|  + JETfw.pkg
|  + JETfwpm.pkg
|  + JETjre.pkg
|  + JETjsvc.pkg
|  + JETm-asrv.pkg
|  + JETm-cron.pkg
|  + JETm-send.pkg
|  + JETmserv.pkg
```

```
|  + JETm-snrt.pkg
|  + JETm-z2c.pkg
|  + JETm-z2.pkg
|  + JETsnort.pkg
|  + JETtw.pkg
|  + JETustat.pkg
|  + keymaker.sh
|  + MD5SUM
|  + packages
|  + pfil.pkg
|
+--Z-2-sparc-sun-solaris2.9/ - Каталог, содержащий инталляционные пакеты
   +                          МЭ "Z-2" для Solaris 9 SPARC
   + BUILD
   + fwinstall.sh
   + fwunattended.sh
   + fwuninstall
   + iengine.sh
   + ipf.pkg
   + JETcfgen.pkg
   + JETfile.pkg
   + JETfilex.pkg
   + JETfw.pkg
   + JETfwpm.pkg
   + JETfwx.pkg
   + JETjre.pkg
   + JETjsvc.pkg
   + JETm-asrv.pkg
   + JETm-cron.pkg
   + JETm-send.pkg
   + JETmserv.pkg
   + JETm-snrt.pkg
   + JETm-z2c.pkg
   + JETm-z2.pkg
   + JETsnort.pkg
   + JETtw.pkg
   + JETustat.pkg
   + keymaker.sh
   + MD5SUM
   + packages
   + pfil.pkg
```

2.5 Установка межсетевого экрана «Z-2»

2.5.1 Порядок установки МЭ «Z-2»

Инсталляция МЭ «Z-2» производится только от имени суперпользователя в однопользовательском режиме операционной системы. Переход в однопользовательский режим необходим для обеспечения недоступности компьютера, на который производится установка ПО, из сети в момент установки.

Порядок установки МЭ «Z-2» следующий:

1. Перезагрузить ОС Solaris в однопользовательский режим. Перевод ОС в однопользовательский режим из многопользовательского должен осуществляться путем перезагрузки компьютера, что обеспечивает корректное завершение всех сетевых сервисов и дополнительных программ и отсутствие загруженных сетевых сервисов на момент установки ПО. Перезагрузка компьютера в однопользовательский режим осуществляется командой:

```
/usr/sbin/reboot -- -s
```

2. Зарегистрироваться в системе от имени суперпользователя с системной консоли.
3. Смонтировать дополнительные файловые системы. После загрузки в однопользовательский режим необходимо осуществить монтирование дополнительных файловых систем командой *mount -a* (т.к. в однопользовательском режиме ОС Solaris не выполняет автоматическое монтирование всех файловых систем).
4. Смонтировать дистрибутивный носитель ПО МЭ «Z-2».
5. На смонтированной файловой системе перейти в каталог *Z-2* и запустить командную процедуру *fwinstall*. Запуск командной процедуры fwinstall осуществляется командой:

```
/bin/sh ./fwinstall.sh
```

Описанная процедура выполняет инсталляцию всех компонент межсетевого экрана, кроме Графического Интерфейса администратора (GUI). Инсталляция графического интерфейса администратора должна производиться на отдельный компьютер (рабочее место администратора межсетевого экрана). Процедура инсталляции графического интерфейса администратора описана в книге *«Универсальный графический интерфейс 2.4 Руководство администратора»*, раздел *«Инсталляция графического интерфейса»*.

2.5.2 Выполнение процедуры инсталляции

Программа установки *fwinstall* является интерактивной — в процессе своей работы выполняет несколько проверок и требует ответить на несколько вопросов.

fwinstall осуществляет следующие проверки:

1. Проверяется номер версии ОС Solaris. Если версия ОС отличается от необходимой, то выполнение процедуры инсталляции завершается с сообщением «You need Solaris 9 OS for install Z-2 verson x.x.x from this disk».
2. Проверяется наличие уже установленного ПО МЭ «Z-2». Если обнаруживается, что ПО МЭ «Z-2» уже установлено, то выполнение процедуры инсталляции завершается с сообщением «You already have Z-2 firewall installed on your system. Please uninstall old version first».
3. Проверяется наличие полномочий суперпользователя у пользователя, выполняющего процедуру инсталляции. Если пользователь не обладает полномочиями суперпользователя, то выполнение процедуры инсталляции завершается с сообщением «You must have superuser access for install Z-2 firewall».

4. Проверяется режим исполнения ОС (runlevel). Если ОС исполняется не в однопользовательском режиме, выдается предупреждающее сообщение.

В процессе установки ПО программа инсталляции задает вопрос — «Do you want server act as alias server? {y/n}?» При ответе «Y» инсталлируемый МЭ будет являться *сервером определений*. Сервер определений хранит информацию об именах хостов, сетей, IP-групп и других сетевых составляющих, заданных администратором (определения). Сервером определений назначается один из используемых в сети МЭ (смотри книгу *«Универсальный графический интерфейс 2.4 Руководство администратора»*).

Далее программа запрашивает о необходимости установить систему обнаружения атак SNORT IDS — «Do you want to install SNORT IDS {y/n}?» При положительном ответе в МЭ будут добавлены компоненты, обеспечивающие интеграцию со Snort IDS.

После этого программа инсталляции предлагает ввести адрес компьютера, с которого будет осуществляться управление МЭ «Z-2». В ответ на запрос процедуры «Enter GUI manager IP address for this firewall (xxx.xxx.xxx.xxx or 'any' or 'none') [any]:» следует ввести IP-адрес в стандартном формате (xxx.xxx.xxx.xxx), определенный на этапе подготовки к инсталляции, или слово «any», или слово «none».

Ввод адреса требуется для того, чтобы программа инсталляции смогла создать правило для пакетного фильтра, позволяющее прохождение пакетов между графическим интерфейсом и сервером управления межсетевого экрана. В случае ответа «any» доступ для управления межсетевым экраном будет разрешен с любого адреса. В случае ответа «none» доступ будет запрещен. При ошибке ввода адреса процедура выдает предупреждающее сообщение и не создает никаких правил фильтрации.

Создаваемое правило является временным и используется только для обеспечения сетевого взаимодействия графического интерфейса и управляющего сервера при первом запуске графического интерфейса. После первого запуска графического интерфейса список правил для пакетного фильтра будет изменен средствами графического интерфейса.

Далее, в процессе инсталляции создается сертификат открытого ключа сервера, который будет использоваться для поддержки целостности и конфиденциальности передаваемых данных.

Для создания сертификата ключа необходимо ввести *определяющее имя* владельца сертификата — выступающее именем сервера.

Для этого необходимо ответить на несколько вопросов, предлагаемых программой инсталляции, или воспользоваться значениями по умолчанию.

Определяющее имя записывается в формате X.500 и состоит из следующих частей:

```
"CN=<имя>,OU=<отд>,O=<орг>,L=<город>,S=<район/штат>,C=<страна>"
```

CN — элемент определяет имя владельца сертификата
OU — элемент определяет отдел организации
O — элемент определяет название организации
L — элемент определяет место расположения владельца сертификата/организации (город, поселок и т.д.)
S — элемент определяет район(область, штат) страны, в котором расположен город
C — элемент определяет название страны

После подтверждения созданного *определяющего имени* необходимо определить логин/пароль администратора для доступа к МЭ.

Далее программа инсталляции предлагает произвести перезагрузку компьютера после установки всех инсталляционных пакетов — «Do you want to reboot after installation process {y/N}?» Перезагрузка необходима для того, чтобы загрузить ядерные модули пакетного фильтра, задействовать обновившуюся конфигурацию и загрузить управляющий сервер МЭ «Z-2».

После перезагрузки системы процедура инсталляции будет завершена.

2.5.3 Автоматическая установка МЭ Z–2

Автоматическая массовая установка МЭ Z-2 осуществляется при помощи специального скрипта-шаблона, который необходимо подготовить перед началом процесса инсталляции. В шаблоне содержатся инструкции и параметры, необходимые для корректной установки Z-2, позволяющие произвести полностью автоматическую инсталляцию программного обеспечения без участия администратора.

Скрип-шаблон генерируется автоматически программой интерактивной установки `fwinstall.sh`. Для создания скрипта-шаблона необходимо перейти в каталог с дистрибутивом Z-2 и запустить fwinstall.sh со следующими параметрами:

```
fwinstall.sh -g
```

— результат работы программы будет помещен в стандартный поток вывода

или

```
fwinstall.sh -g -f  <имя файла>
```

— результат работы программы будет помещен в файл с заданным именем.

Программа `fwinstall.sh` запустится в режиме имитации установки программного обеспечения, без реальной инсталляции файлов. После ответов на все вопросы скрипт-шаблон будет выдан в стандартный поток вывода или в файл с заданным именем.

При массовой установке МЭ Z-2 для удобства можно отредактировать содержимое созданного файла в соответствии с потребностями конкретной конфигурации. Все параметры в файлах примеров имеют комментарии, поэтому сложностей с их адаптацией к конкретным условиям быть не должно.

Для автоматической инсталляции Z-2 необходимо:

1. на целевую машину, во временный каталог перенести дистрибутив и подготовленный скрипт-шаблон;
2. перейти в каталог, куда был перенесен дистрибутив
3. запустить скрипт массовой установки Z-2:

```
./fwunattended.sh -a
```

Примечание При запуске `fwunattended.sh` без параметров выдается помощь. При запуске программы с параметром `-a` осуществляется автоматическая установка Z-2.

Некоторые параметры, которые были сгенерированы при создании файла-шаблона, можно изменить при помощи переменных окружения. Для изменения значений, заданных по умолчанию можно использовать следующие переменные.

INSTALL_ASRV=1 — Установка сервера определений.
INSTALL_SNORT=0 — Установка SNORT.
DO_REBOOT=0 — Перезагрузиться после установки.
USE_SKEY=0 — Сконфигурировать S/Key
GUI_RULE=any — Создание правила по умолчанию для GUI (none, any or dotted IP).
DISTRIBUTIVE_DIR — Местоположение пакетов (по умолчанию — текущий каталог).
SKEY_DN — DN для создания S/key
KEYCONFIG — Скрипт для конфигурации S/key

POSTINSTALL_SCRIPT, PREINSTALL_SCRIPT — Скрипты, вызываемые автоматически до и после установки. По умолчанию поиск скриптов `preinstall.sh` и `postinstall.sh` происходит в текущем каталоге. В случае, когда выполнение `preinstall.sh` завершается с ошибкой, установка прерывается.

FW_DNAME — Содержит полный DNAME

2.5.4 Деинсталляция МЭ «Z-2»

Деинсталляция Z-2 производится при помощи программы Z-2 `fwuninstall`. Версия программы `fwuninstall` должна соответствовать версии МЭ Z-2. При переинсталляции МЭ Z-2 необходимо провести полную деинсталляцию предыдущей версии.

Процедура деинсталляции изменяет следующие системные файлы:

/etc/syslog.conf
возвращает сохраненный перед инсталляцией файл

/etc/pam.conf
возвращает сохраненный перед инсталляцией файл

/etc/mail/aliases
возвращает сохраненный перед инсталляцией файл

Для деинсталляции необходимо провести следующие действия.

1. Запустить программу деинсталляции МЭ `fwuninstall` с параметром `-r`. Для удаления продукта без вопросов служит дополнительный параметр `-f`. Например:

   ```
   #  /opt/fw/sbin/fwuninstall -r -f
   ```

   При запуске без параметров выдается справка об использовании программы.
   Примечание Не следует использовать программу деинсталляции с дистрибутивного компакт-диска
2. Удалить каталоги с оставшимися после деинсталляции файлами:

   ```
   # rm -rf /opt/fw
   # rm -rf /opt/JETmserv
   # rm -rf /etc/opt/fw
   # rm -rf /var/spool/z2
   # rm -rf /var/spool/z2bad
   ```

3. Проверить crontab на предмет относящихся к Z-2 записей:

   ```
   # EDITOR=vi crontab -e
   ```

Примечание Программа деинсталляции Z-2 версии 1.8 расположена в каталоге /opt/fw/bin

Примечание Программа деинсталляции Z-2 версии 1.6 не устанавливается в систему, необходимо использовать деинсталлятор с дистрибутивного компакт-диска.

Возможные проблемы:

1. При использовании деинсталлятора с дистрибутивного диска для Z-2 версий 1.8/2.0 не удаляются опционально установленные пакеты JETm-asrv и JETcron. Для решения этой проблемы необходимо удалить эти пакеты командой pkgrm.

2. При переустановке Z-2 без удаления предыдущей версии программы или при удалении Z-2 деинсталлятором от другой версии программы последующая установка Z-2 может пройти некорректно — часть пакетов может быть пропущена при установке, некоторые системные файлы могут быть некорректно сконфигурированы (главным образом pam.conf), некоторые файлы могут вызывать конфликты из-за несоответствия версий. Для решения этой проблемы необходимо произвести переинсталляцию ОС Solaris и установить Z-2 с нуля.

2.5.5 Обновление МЭ Z–2 версий 1.8/2.0/2.2 до Z–2 версии 2.4

Процедура обновления МЭ Z-2 версий 1.8/2.0/2.2 до Z-2 версии 2.4 состоит из следующих этапов:

1. Сохранение конфигурации установленной версии МЭ
2. Деинсталляция старой версии МЭ Z-2
3. Установка Z-2 версии 2.4
4. Восстановление конфигурации сохраненной версии МЭ

2.5.5.1 Сохранение конфигурации установленной версии МЭ

Для сохранения конфигурации МЭ используется утилита save_config.sh, которая находится в каталоге /opt/fw/sbin (подробнее см. раздел 12.1.1 на стр. 159). Данная утилита сохраняет конфигурацию в виде tar-архива.

После запуска утилиты необходимо удостовериться, что в архив попали следующие файлы:

- `firewall.xml` — конфигурация МЭ (для всех конфигурационных наборов);
- `mserv.xml` — конфигурация MSERV;
- `aliases.xml` для версии Z-2 1.8, каталог `aliases.d` для версии Z-2 2.0/2.2 — конфигурация `Сервера определений`;
- базы данный пользователей из /var/lib/z2 (при использовании аутентификации);
- `snort.xml` — конфигурация SNORT (при использовании snort).

2.5.5.2 Деинсталляция старой версии МЭ Z–2

Для корректной установки новой версии МЭ Z-2 необходимо провести деинсталляцию старой версии программы. Из системы должны быть удалены все пакеты старой версии МЭ Z-2.

Процедура деинсталляции описана в разделе 2.5.4 на стр. 32 данного руководства.

2.5.5.3 Установка Z–2 версии 2.4

Процедура установки межсетевого экрана «Z-2» описана в разделе 2.5 на стр. 29 данного руководства.

2.5.5.4 Восстановление конфигурации сохраненной версии МЭ Z–2

2.5.5.4.1 Восстановление конфигурации `mserv` и `Сервера определений`

Конфигурация mserv (файл `mserv.xml`) восстанавливается вручную, через GUI. Для этого необходимо в разделе `Сервер управления` Графического интерфейса выставить соответствующие значения, взяв их из сохраненного файла `mserv.xml` ранней версии Z-2.

Набор определений `Сервера определений` можно не восстанавливать — МЭ Z-2 версий 1.8/2.0/2.2 сохраняет полную копию `Сервера определений` в файле `firewall.xml`. Восстановление сервера определений происходит одновременно с восстановлением конфигурации МЭ (см. раздел 2.5.5.4.2 на стр. 34).

2.5.5.4.2 Восстановление конфигурации МЭ

Графический интерфейс МЭ Z-2 2.4 может загружать предыдущие версии конфигурационных файлов. Таким образом, возможны два способа восстановления конфигурации.

1. Установка сохраненной копии файла `firewall.xml` в каталог /opt/JETmserv/etc/current взамен файла конфигурации по умолчанию.
2. Импортирование сохраненной конфигурации из GUI как нового конфигурационного набора[1].

Примечание Также можно явно преобразовать старый `firewall.xml` в новый формат при помощи консольной утилиты `upgrade-config`, находящейся в каталоге с GUI. Синтаксис команды следующий:

upgrade-config старый_firewall.xml новый_firewall.xml

Использование старой конфигурации МЭ возможно без запуска GUI — с помощью программы генерации конфигурации `confgen`. Программа `confgen` создает новую конфигурацию прикладных шлюзов и пакетного фильтра из сохраненной копии файла `firewall.xml`. Однако, при первом соединении с GUI произойдет восстановление `Сервера определений`. В случае, если в ходе инсталляции изменились сетевые интерфейсы, GUI предложит задать схему их переопределения.

2.5.6 Обновление дистрибутива в рамках одной версии МЭ Z–2

Исполняемые модули МЭ Z-2 собраны в несколько пакетов ОС Solaris. Пакеты разработаны таким образом, чтобы их удаление не затрагивало созданных пользователем конфигурационных файлов. Таким образом, обновление какой-либо компоненты МЭ «Z-2» производится путем удаления соответствующего ему пакета и последующей инсталляцией его обновления. Конфигурационные файлы дополнительно сохранять не требуется.

Тем не менее, перед обновлением рекомендуется произвести полное сохранение конфигурации МЭ с помощью скрипта `save_config.sh` (подробнее см. раздел 12.1.1 на стр. 159).

Примечание Таким способом можно, например, обновить МЭ Z-2 версии 2.4.0 до версии 2.4.1. Но нельзя обновить МЭ Z-2 версии 1.8/2.2 до версии версии 2.4.

2.5.6.1 Обновление серверной части: прикладные шлюзы

Прикладные шлюзы расположены в пакетах JETfw и JETfwx (последнее - только на Sparc). Для обновления прикладных шлюзов необходимо:

1. Скопировать обновление на сервер
2. Остановить все прикладные шлюзы командой:
   `# /opt/JETmserv/sbin/fwctl stopall`
3. Удалить пакеты командами:
   `# pkgrm JETfw`
   - для всех платформ
   `# pkgrm JETfwx`

[1]Раздел Графического интерфейса `Конфигурация МЭ`/кнопка «Загрузить конфигурацию МЭ»

- только для Sparc

4. Установить новые пакеты:

   ```
   # pkgadd -d <путь>/JETfw.pkg
   ```

   - для всех платформ

   ```
   # pkgadd -d <путь>/JETfwx.pkg
   ```

   - только для Sparc

5. Запустить прикладные шлюзы:

   ```
   # /opt/JETmserv/sbin/fwctl startall
   ```

2.5.6.2 Обновление серверной части: модули управления и генератор конфигурации

Модули управления расположены в пакетах: JETmserv, JETm-cron, JETm-z2, JETm-z2c, JETcfgen (генератор конфигурации), JETjre (Java-runtime), JETustat (библиотека доступа к функциям ядра Solaris), JETjsvc (среда запуска сервера управления). Для обновления любого из перечисленных компонентов необходимо:

1. Скопировать обновление на сервер.
2. Закрыть все соединения GUI к серверу.
3. Остановить сервер управления (mserv) командой:

   ```
   # /opt/JETmserv/sbin/mserv stop
   ```

4. Удалить соответствующий пакет(ы) командой:

   ```
   # pkgrm <имя пакета>
   ```

5. Установить новый(е) пакет(ы):

   ```
   # pkgadd -d <путь>/<имя пакета.pkg>
   ```

6. Запустить сервер управления (mserv) командой:

   ```
   # /opt/JETmserv/sbin/mserv start
   ```

Глава 3

Архитектура межсетевого экрана «Z-2»

МЭ «Z-2» состоит из следующих компонент:

Пакетный фильтр — осуществляет фильтрацию пакетов на транспортном и сетевом уровнях, контроль открытых через МЭ соединений. Трансляцию сетевых адресов. Учет трафика на сетевом уровне. Поддержку *прозрачного* режима работы прикладных шлюзов. Подробно пакетный фильтр описан в главе 5 на стр. 47.

Прикладные шлюзы — обеспечивают прохождение соединений через МЭ с контролем прикладного протокола и фильтрацией на основе адресов и других параметров прикладного уровня. Подробно прикладные шлюзы описаны в главе 6 на стр. 69.

Подсистема восстановления — осуществляет функции сохранения и восстановления конфигурации МЭ.

Подсистема контроля целостности — осуществляет контроль изменений программных компонентов МЭ «Z-2».

Сервер аутентификации — обрабатывает запросы на аутентификацию пользователей от прикладных шлюзов. Обеспечивает работу прикладных шлюзов с аутентификационными базами данных сторонних производителей по PAM протоколу. Подробно сервер аутентификации описан в главе 7 на стр. 129.

Подсистема анализа регистрационной информации — предназначена для анализа и составления отчетов на основе записей компонент МЭ «Z-2» в системные журналы.

Управляющий сервер — предоставляет модули управления МЭ через графический интерфейс.

Графический интерфейс администратора — предназначен для управления МЭ. Интерфейс устанавливается на отдельное АРМ администратора и управляет МЭ по сети.

3.1 Прохождение пакетов через МЭ «Z-2»

Пакет проходит следующие подсистемы МЭ «Z-2» при его маршрутизации:

1. *Сетевая трансляция адресов на входе интерфейса*. На этом этапе осуществляется трансляция адреса назначения пакета. Это используется при работе шлюзов в прозрачном режиме и переадресации. Пакет с транслированным адресом назначения попадает в подсистему маршрутизации ОС Solaris, минуя дальнейшие проверки пакетного фильтра. Подробно этот этап описан в главе 5.2 на стр. 51.
2. *Проверка пакета по входящим правилам пакетного фильтра*. Осуществляется проверка пакета по правилам пакетного фильтра. На основе проверки пакет может быть пропущен, отброшен либо отброшен с уведомлением отправителю. Подробнее этот этап описан в главе 5 на стр. 47.
3. *Системная маршрутизация*. Осуществляется ядром операционной системы Solaris. На основе таблицы маршрутизации пакет может быть направлен на исходящий интерфейс либо на процесс прикладного шлюза. Решение производится на основе адреса назначения пакета.
4. *Проверка пакета по исходящим правилам пакетного фильтра*.
5. *Сетевая трансляция адресов на выходе интерфейса*. На этом этапе осуществляется трансляция исходящего адреса пакета.

Пакеты, исходящие с процессов МЭ (например, с прикладных шлюзов), попадают непосредственно на стадию системной маршрутизации.

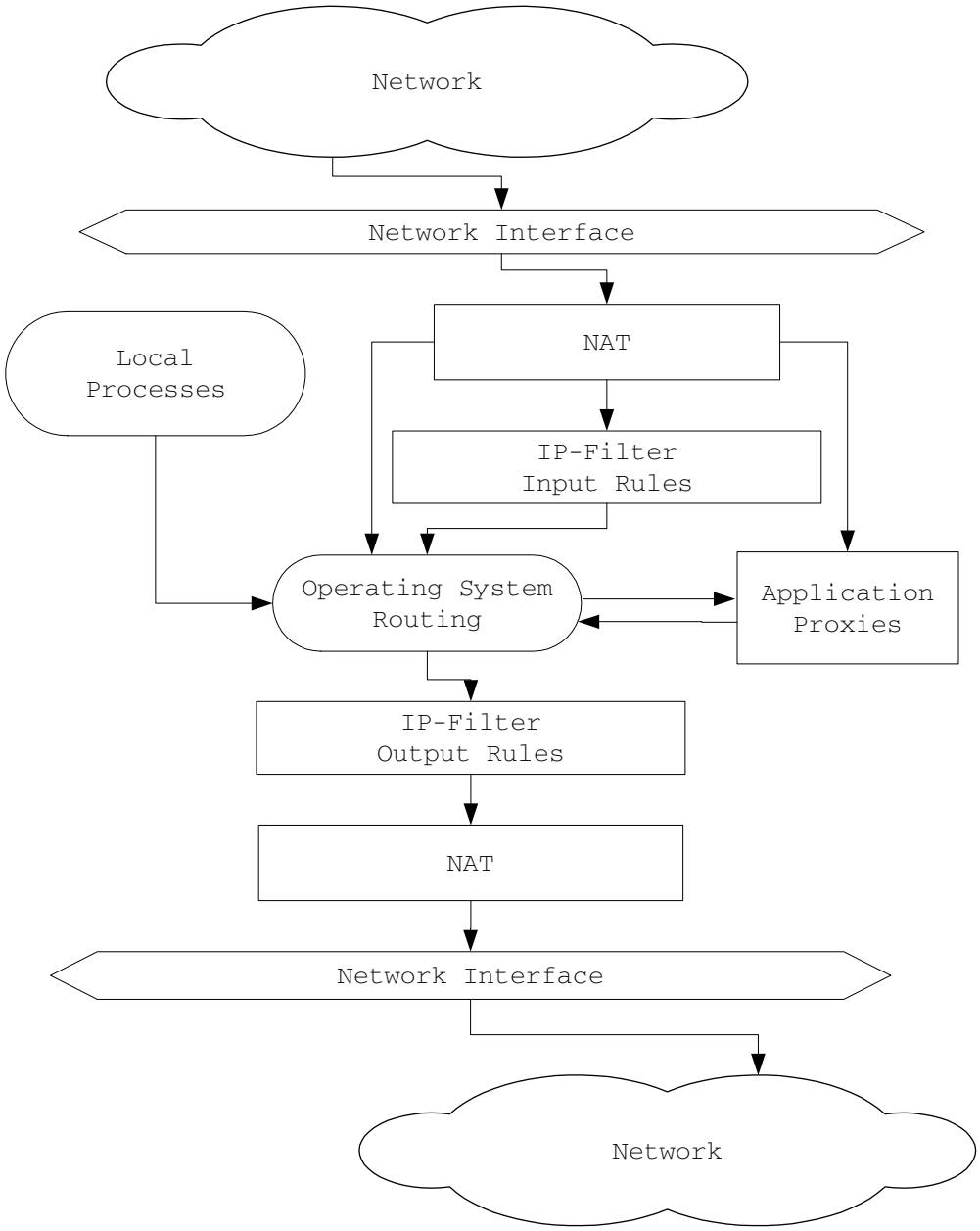

Рис. 3.1: Диаграмма прохождения пакета через МЭ «Z-2»

3.2 Варианты фильтрации соединений

При конфигурации МЭ существуют следующие варианты фильтрации соединений через МЭ «Z-2»:

- через пакетный фильтр;
- через специализированный прикладной шлюз;
- через прикладной шлюз общего назначения.

Администратор межсетевого экрана должен выбрать вариант, наиболее подходящий для реализации выбранной им политики безопасности.

Пакетный фильтр является наиболее производительным способом пропускания пакета. Недостатком пакетного фильтра является его принципиальная неспособность к анализу передаваемых в пакете данных и количеству открытых сессий. Также следует отметить, что существует ряд протоколов прикладного уровня (FTP, Oracle SQL*NET, RealAudio/Video и т.д.), передающих информацию сетевого уровня в контексте протокола, что делает возможность поддержки таких протоколов только средствами пакетного фильтра весьма проблематичной.

Специализированный прикладной шлюз производит контроль прикладного протокола, дает возможность фильтрации на основе прикладных адресов (и других прикладных критериев), а также осуществляет журнализацию прикладных адресов, сессий и команд протокола. Соединения пользователей поступают на прикладной шлюз, далее шлюз от своего имени осуществляет запрос пользовательского ресурса. Благодаря этому, прикладные шлюзы делают невозможными ряд атак, использующих особенности реализации транспортного и сетевого уровня. Прикладные шлюзы также позволяют ограничить количество одновременно открытых сессий обслуживаемого ими протокола. Основным недостатком прикладных шлюзов является увеличение времени отклика, вносимое работой прикладного шлюза. Нагрузка на сервер МЭ у прикладных шлюзов значительно выше пакетного фильтра.

Прикладные шлюзы общего назначения не осуществляют контроль прикладного протокола. Эти прикладные шлюзы обеспечивают реконструкцию соединения и таким образом обеспечивают защиту от атак транспортного и сетевого уровней. Прикладные шлюзы, как и специализированные, позволяют ограничить количество одновременно открытых через них сессий. Недостатками таких шлюзов являются вносимые ими задержки, а также невозможность анализа передаваемых данных.

В таблицах показаны рекомендуемые варианты фильтрации в типовых конфигурациях. Знаком «+» обозначен рекомендуемый вариант, «x» отмечены невозможные варианты. Пустая ячейка обозначает возможный, но не рекомендованный вариант.

Таблица 3.1: **Доступ пользователей к ресурсам интернет. Рекомендуемые варианты фильтрации**

протокол	**пакетный фильтр**	**специализированный шлюз**	**шлюз общего назначения**
HTTP		+	
FTP	x	+	x
TELNET	+	+	+
SMTP		+	
POP	+	+	+
IMAP/NNTP	+	x	+
DNS	+	x	+

Таблица 3.2: **Доступ внешних пользователей к ресурсам внутренней сети. Рекомендуемые варианты фильтрации**

Продолжение следует

. . . продолжение таблицы 3.2.

протокол	**пакетный фильтр**	**специализированный шлюз**	**шлюз общего назначения**
HTTP		+	+
FTP	x	+	x
TELNET		+	+
SMTP		+	
POP/IMAP/NNTP		x	+
DNS	+	x	+

Глава 4

Демонстрационный пример

В дальнейшем, для описания настроек пакетного фильтра и прикладных шлюзов будет использоваться типичная схема включения межсетевого экрана в корпоративную сеть, приведенная на рис. 4.1.

Предположим, что корпоративная сеть имеет выделенный канал для подключения к сети провайдера, и провайдер выделил 8 «реальных» («маршрутизируемых») адресов (208.172.16.0 — 208.172.16.7 из сети 208.172.16.0/27) для публичных сервисов из своего адресного пространства и зарегистрировал домен *my-domain.ru*.

В таблице 4.1 приведены адреса и имена публичных сервисов компании.

Таблица 4.1: **Распределение выделенных маршрутизируемых адресов**

адрес	**имя сервиса**	**комментарий**
208.172.16.1	rtr-int	Адрес внутреннего интерфейса маршрутизатора
208.172.16.2	firewall	Адрес интерфейса qfe0 МЭ «Z-2»
208.172.16.3	www.my-domain.ru	Корпоративный веб-сервер
208.172.16.4	ftp.my-domain.ru	Корпоративный FTP-сервер
208.172.16.5	appl-server.my-domain.ru	Корпоративный сервер приложений

Компьютер, на который установлено ПО МЭ «Z-2», имеет несколько сетевых интерфейсов. Для определенности, пусть на нем установлен четырехпортовый адаптер QFE (quad-fast-ethernet). Интерфейс qfe0 соединен с маршрутизатором и имеет адрес 208.172.16.2.

Интерфейс qfe1 соединен с отдельным сегментом сети, в котором размещены серверы, предоставляющие публичные сервисы. Сегмент, в котором размещаются публичные сервисы (доступные извне), обычно отделяется от остальных сегментов корпоративной локальной сети как физически, так и на уровне маршрутизации. По традиции, сегмент, в котором размещены публичные серверы называется «демилитаризованной зоной» (*DMZ*). Основным ограничением в конфигурации МЭ при организации *DMZ* является запрет на установку сетевых соединений из *DMZ* во внутреннюю сеть.

Серверы, расположенные в сегменте *DMZ*, имеют локальные адреса из диапазона адресов сети класса «C» 192.168.10.0/24. Трансляция локальных адресов производится средствами МЭ «Z-2».

Интерфейс qfe2 соединен с сегментом, образующим внутреннюю (пользовательскую) локальную сеть. Все компьютеры, включенные во внутреннюю сеть, также имеют локальные адреса из диапазона адресов сети класса «C» 172.16.10.0/24. Доступ во внешнюю сеть (Интернет) для пользователей внутренней сети осуществляется посредством трансляции адресов, выполняемой МЭ «Z-2».

Рассмотрим теперь конфигурацию корпоративной сети с точки зрения предоставляемых ею сервисов.

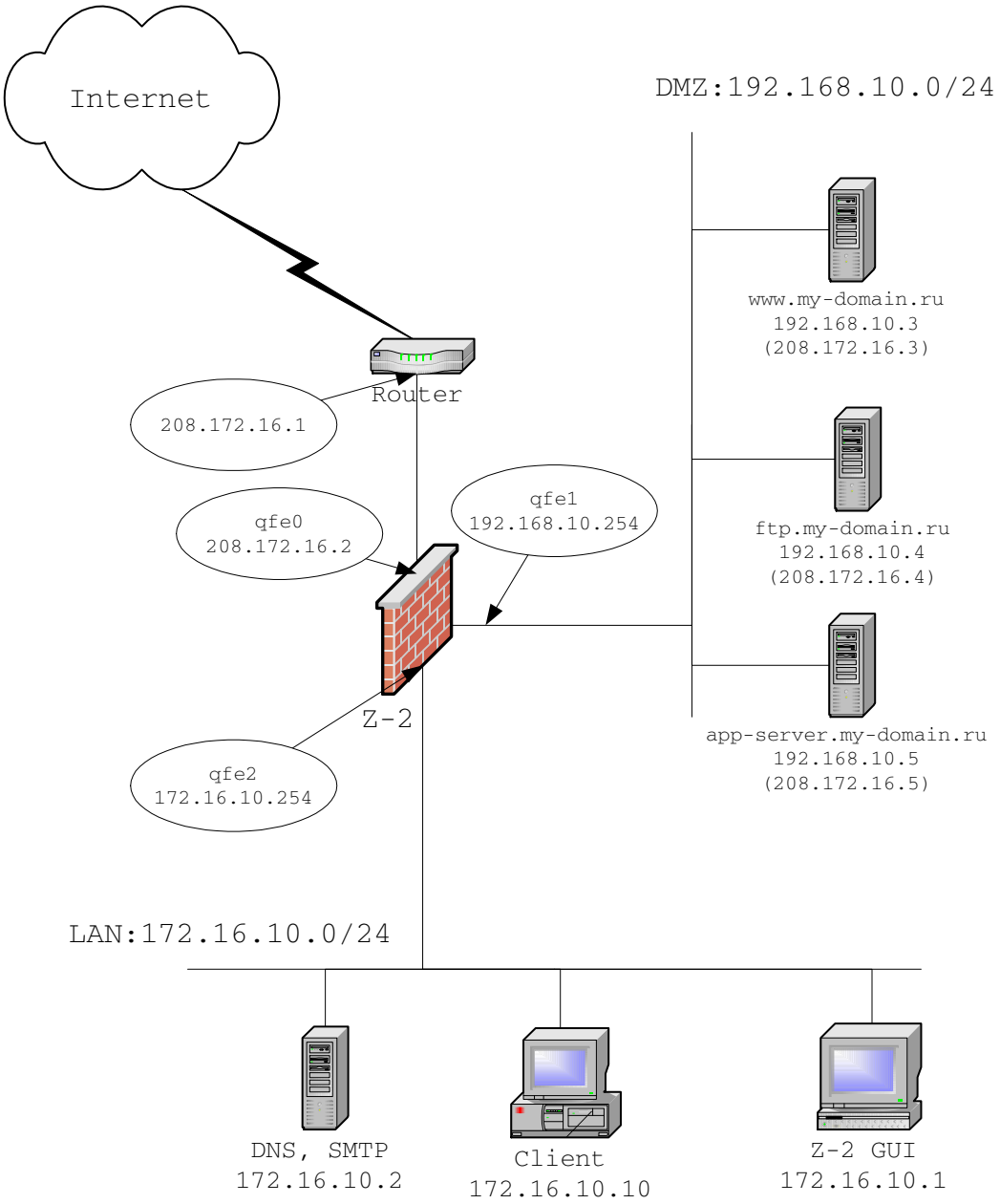

Рис. 4.1: Схема подключения корпоративной сети

В таблице 4.2 перечислены все доступные сетевые сервисы общего пользования вместе с именами и адресами серверов, предоставляющих указанный сервис.

Таблица 4.2: **Публичные сервисы корпоративной сети**

Сервис	**Имя сервера**	**Адрес/интерфейс**	**Комментарий**
Первичный сервер DNS для «внешнего мира»	Z-2	208.172.16.2 (qfe0)	Первичный DNS-сервер для зоны mydomain.ru. Предназначен для использования из внешней сети. Сервер должен ожидать запросы к себе только на интерфейсе qfe0 и содержать в своей базе данных только имена и адреса публично-доступных серверов.
Первичный сервер DNS для внутренней сети.	Z-2	172.16.10.254 (qfe2)	Первичный DNS-сервер для зоны mydomain.ru. Предназначен для использования из внутренней сети. Сервер должен ожидать запросы к себе только на интерфейсе qfe1. В своей базе данных сервер содержит адреса/имена как внешних так и внутренних узлов корпоративной сети.
Корпоративный WEB-сервер	www	192.168.10.3 (208.172.16.3)	Корпоративный WEB-сервер, доступный как из внешней сети (Интернет), так и из внутренней сети. Размешен в DMZ.
Корпоративный FTP-сервер	ftp	192.168.10.4 (208.172.16.4)	Корпоративный FTP-сервер, доступный как из внешней сети (Интернет), так и из внутренней сети. Размешен в DMZ.
Корпоративный сервер приложений	app-server	192.168.10.5 (208.172.16.5)	Корпоративный сервер приложений, доступный как из внешней сети (Интернет), так и из внутренней сети. Размешен в DMZ. Доступ осуществляется протоколом TELNET.
Корпоративный почтовый сервер	smtp	172.16.10.2	Корпоративный почтовый сервер. Размещен во внутренней сети. Также выполняет функции вторичного DNS-сервера для внутренней сети.

Глава 5

Пакетный фильтр

Пакетный фильтр (`ipfilter`) представляет собой отдельную компоненту МЭ «Z-2», предназначенную для разграничения и фильтрации трафика между внешними и внутренними сетями на сетевом и транспортном уровне. Использование модуля `ipfilter` позволяет:

1. Выделять и анализировать IP-пакеты по адресу/порту отправителя.
2. Выделять и анализировать IP-пакеты по адресу/порту получателя.
3. Выделять и анализировать IP-пакеты по номеру протокола, инкапсулированного в IP-пакет.
4. Выделять и анализировать IP-пакеты по имени интерфейса, на который получен пакет.
5. Выделять и анализировать фрагментированные IP-пакеты и обрабатывать фрагменты.
6. Анализировать содержимое служебных полей (флагов) заголовка IP-пакета и выделять IP-пакеты по значению флагов.
7. Выделять и анализировать IP-пакеты по «направлению движения», то есть отдельно выделять входящие в сетевой интерфейс и исходящие из него пакеты.
8. Производить сетевую трансляцию адресов.
9. Поддерживать работу прикладных шлюзов в *прозрачном* режиме.
10. Отслеживать открытые через МЭ `TCP`-соединения и *псевдо-сессии* для протоколов `UDP` и `ICMP`.
11. Подсчитывать объем переданного и полученного трафика для целей сбора статистики и измерения загрузки канала передачи данных.

Правила пакетного фильтра задаются в разделе `Конфигурация МЭ -> Пакетный фильтр -> Контроль доступа` графического интерфейса (рисунок 5.1) и обрабатываются подсистемой фильтрации сверху вниз до первого совпадения.

Выделенный по некоторому критерию пакет может быть:

1. Отправлен получателю (действие `packet permit`).
2. Отправлен получателю с сохранением состояния соединения (действие `session permit`).
3. Отправлен получателю и подсчитан (действие `count`).
4. Заблокирован (действие `deny`).
5. Заблокирован с уведомлением отправителю (действие `reject`).

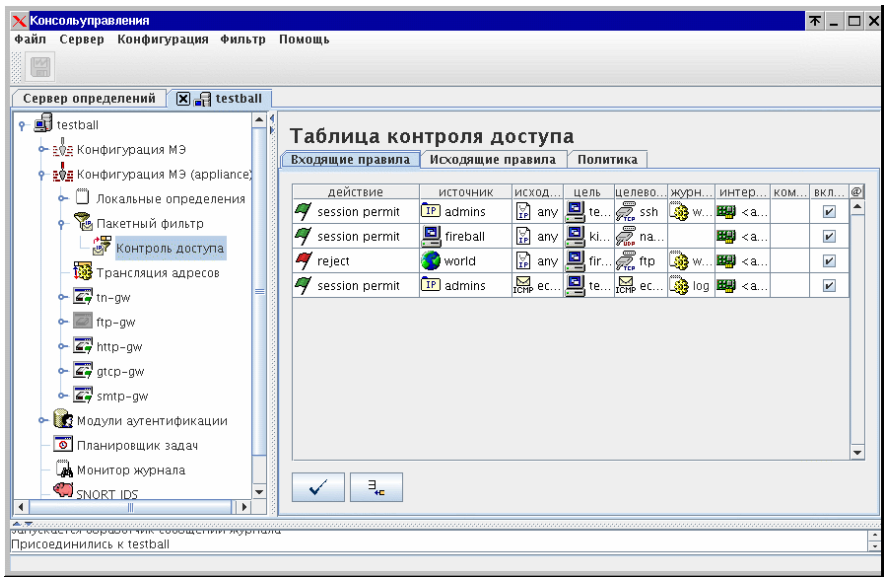

Рис. 5.1: Правила трансляции адресов в графическом интерфейсе

5.1 Фильтрация с сохранением состояния

Отличительной особенностью пакетного фильтра является его способность отслеживать установленные через него соединения для заданных администратором правил фильтрации, то есть сохранять состояние соединения. Режим «сохранение состояния» задается в каждом правиле указанием `session permit`, и поддерживается только для протоколов TCP,UDP или ICMP.

Принцип фильтрации с сохранением состояния состоит в следующем:

- Для первого пакета, соответствующего правилу, создается запись в *таблице состояний* пакетного фильтра. В этой таблице хранится информация об установленной сессии — адрес и порт отправителя, адрес и порт получателя, а также различная служебная информация в зависимости от используемого протокола.
- Любой пакет, проходящий через МЭ, проверяется на принадлежность записям *таблице состояний*. Если пакет соответствует какой-либо записи, то есть принадлежит ранее установленной сессии, то он пропускается через МЭ без дальнейших проверок.
- Если пакет не принадлежит ни одной из ранее установленных сессий, то проходит проверку на соответствие входным правилам фильтрации.

Такая технология фильтрации имеет ряд неоспоримых преимуществ. Во-первых, использование фильтрации с сохранением состояния значительно повышает безопасность и удобство работы с межсетевым экраном. Предположим, что нам необходимо разрешить доступ с машины «client» на TELNET-сервер, работающий на машине «server».

Без использования фильтрации с сохранением состояния нам необходимо создать два правила:

1. Правило, разрешающее доступ с «client» на «server» на порт 23 по протоколу TCP с возможностью открытия соединения
2. Правило, разрешающее доступ с «server» с порта 23 на «client» на любой порт, больший 1024, без возможности открытия соединения

Кроме того, что это не очень удобно с точки зрения администратора, данная комбинация правил неидеальна с точки зрения безопасности управления доступом, так как разрешает прохождение пакетов с машины «server» на машину «client», даже если соединение с машины «client» реально не было открыто.

При использовании фильтрации с сохранением состояния на МЭ необходимо создать только одно правило:

1. Правило, разрешающее доступ с «client» на «server» на порт = 23 для пакета протокола TCP

Кроме этого, при использовании фильтрации с сохранением состояния значительно увеличивается скорость обработки пакетов, так как проверка пакетов в сессии на соответствие заданной политике безопасности производится только для первого пакета, инициализирующего сессию. После этого все пакеты в сессии проверяются только на принадлежность к уже установленной сессии, что выполняется гораздо быстрее, чем проверка на соответствие правилам фильтрации, заданных администратором МЭ.

Рассмотрим особенности работы фильтрации с сохранением состояния для протоколов TCP, UDP и ICMP:

- Для протокола TCP реализация фильтрации с сохранением состояния достаточно тривиальна, так как протокол TCP сам по себе предполагает наличие сессии. Пакетный фильтр сохраняет параметры TCP-сессии в *таблице состояний* и проверяет приходящие пакеты на принадлежность установленным TCP-сессиям.
- Для протокола UDP понятия сессии не существует, поэтому пакетный фильтр эмулирует сессию для UDP-протокола. Первый UDP-пакет открывает сессию, остальные UDP-пакеты считаются

принадлежащими сессии, если они отправлены на IP-адрес и порт источника, открывшего сессию пакета, с IP-адреса и порта его назначения.

- Для протокола ICMP также производится эмуляция сессии. При этом подразумевается, что обмен ICMP-сообщениями производится по схеме *запрос-ответ*, т.е. соответствующий правилу пакет открывает *псевдо-сессию*, которая существует до получения ответного пакета, который эту сессию завершает. Ответный пакет считается принадлежащим сессии, если он имеет соответствующий тип (например, если *псевдо-сессию* инициировал ICMP-пакет с типом *ECHO REQUEST*, то принадлежащий *псевдо-сессии* пакет должен иметь тип *ECHO REPLY*) и исходит с IP-адреса, на который направлялся ICMP-пакет, открывший *псевдо-сессию*.

Таким образом, последовательность обработки пакета на соответствие правилам полностью выглядит следующим образом:

1. Производится проверка пакета на принадлежность к уже установленной сессии.
2. Если пакет принадлежит к уже установленной сессии, то он передается стеку протоколов без проверок на соответствие входным правилам фильтрации.
3. Пакет проверяется на соответствие входному набору правил.
4. Пакет передается выше по стеку протоколов.
5. При получении пакета от стека протоколов на отправку производится проверка пакета на принадлежность к уже установленной сессии.
6. Если пакет принадлежит транспортной сессии, то он передается сетевому интерфейсу для отправки.
7. Иначе, пакет проверяется на соответствие выходному набору правил.

5.2 Сетевая трансляция адресов

Сетевая трансляция адресов (network address translation, *NAT*) решает следующие типовые задачи:

- Сокрытие структуры сети при подключении к сети Интернет.
- Обеспечение взаимодействия внутренней сети, имеющей адреса из зарезервированного блока, с сетью Интернет.
- Обеспечение «прозрачного» режима работы прикладных шлюзов.

Эти задачи также могут быть решены при помощи *прикладных шлюзов*, но есть ряд моментов, которые ограничивают их применение:

- Прикладной шлюз предназначен для обслуживания единственного протокола, для которого он был разработан. Если необходимо обеспечить взаимодействие по протоколу, для которого нет прикладного шлюза, необходима разработка нового прикладного шлюза.
- Прикладные шлюзы, в отличие от пакетной фильтрации, гораздо более ресурсоемки.

В тех случаях, когда применение прикладных шлюзов затруднительно, применяется технологии пакетной фильтрации и сетевой трансляции адресов.

В МЭ «Z-2» реализованы следующие схемы трансляции сетевых адресов:

Динамическая сетевая трансляция адресов (Dynamic NAT)
: трансляция исходного диапазона IP-адресов на другой, меньший диапазон. Подробно этот режим NAT описан в разделе 5.2.1 на стр. 51.

Статическая сетевая трансляция адресов (Static NAT)
: установка двусторонней адресной трансляции между внутренним и внешним IP-адресом по схеме «один к одному». Подробно этот режим NAT описан в разделе 5.2.2 на стр. 55.

Переадресация --- прозрачный шлюз (Redirection)
: режим замены адреса назначения IP-пакета. Правила этого типа обеспечивают функционирование прикладных шлюзов в прозрачном режиме. Подробно этот режим NAT описан в разделе 5.2.3 на стр. 57

Шлюз уровня ядра (Kernel Proxy)
: прикладные шлюзы ядерного уровня. Правило этого типа обеспечивают корректную обработку протоколов, не реализуемых штатными средствами пакетного фильтра. Подробно этот режим описан в разделе 5.2.4 на стр. 58.

Правила трансляции адресов задаются в разделе `Трансляция адресов` графического интерфейса (рисунок 5.2) и обрабатываются подсистемой трансляции сверху вниз до первого совпадения.

Таким образом, порядок размещения правил трансляции адресов важен. Типичным примером является совместное применение правил типа `Динамическая трансляция адресов` и `Шлюз уровня ядра`. Если правила `Динамической трансляции` размещаются перед правилами `Шлюза уровня ядра`, то протоколы, которые должны обрабатываться прикладными шлюзами ядерного уровня, будут просто транслироваться правилами `Динамической трансляции` и работать не будут. Чтобы избежать подобной ситуации, правила `Шлюза уровня ядра` должны размещаться *перед* правилами `Динамической трансляции`. В этом случае нужные протоколы будут обработаны прикладными шлюзами ядерного уровня, а остальные пакеты — правилами динамической трансляции адресов.

5.2.1 Динамическая сетевая трансляция адресов

Динамическая трансляция адресов или `Dynamic NAT` позволяет транслировать пул адресов за межсетевым экраном в один или несколько адресов, как правило, адрес внешнего интерфейса МЭ[1].

[1]При динамической трансляции в пул адресов необходимо обеспечить маршрутизацию этих адресов через МЭ. Варианты решения данной задачи описаны в разделе 5.2.5 на стр. 59.

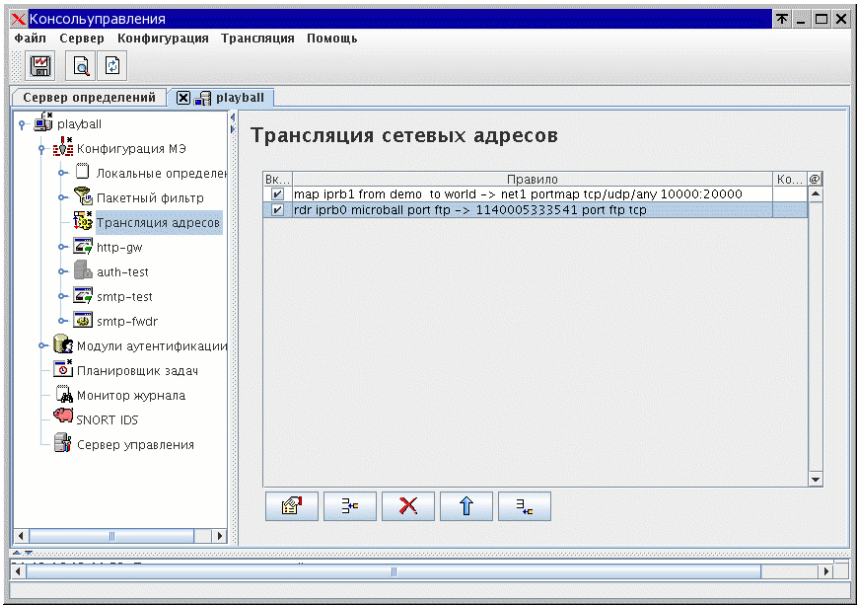

Рис. 5.2: Правила трансляции адресов в графическом интерфейсе

Правила такого типа используются для обеспечения доступа к сети Интернет внутренних сетей организации, имеющих адреса из так называемого «зарезервированного» блока IP-адресов[1].

При применении динамической трансляции есть несколько ограничений:

- При применении правил `Динамической сетевой трансляции адресов` доступ к внутренним (транслируемым) адресам извне невозможен.
- Режим динамической трансляции адресов не может использоваться для протоколов, требующих того, чтобы порт отправителя оставался неизменным. Типичным примером этого является протокол FTP, и для работы с этим протоколом должен использоваться прикладной шлюз или режим `Шлюз уровня ядра`.
- Режим динамической трансляции адресов не может использоваться в тех случаях, когда внешний сервер должен различать клиентов по IP-адресу (это будет невозможно, так как все соединения будут производиться от одного адреса).

Правило динамической трансляция адресов в графическом интерфейсе имеет следующий формат:

```
map <Интерфейс> from <Источник> to <Назначение> ->
<Новый источник> [portmap tcp/udp port1:port2]
```

где:

<Интерфейс>
Имя интерфейса, на котором будет выполняться трансляция адресов. Динамическая трансляция должна производиться на исходящем интерфейсе.

<Источник>
Адрес сети, для которой будет производиться трансляция адресов.

<Назначение>
Адрес сети, при обращении к которой будет производиться трансляция адресов.

<Новый источник>
Адрес, в который будут транслироваться адреса из сети <src_net>

port1:port2
Диапазон портов `Сетевой трансляции адресов` для протоколов TCP и UDP.

Например, если мы хотим, чтобы все адреса из сети *net1* (172.16.0.0/16) при передаче через межсетевой экран преобразовывались в адрес внешнего интерфейса iprb1 ballorian, мы должны создать следующее правило динамической трансляция адресов (рисунок 5.3):

```
map iprb1 from net1 to world -> @ballorian-iprb1 portmap tcp/udp/any
10000:20000
```

При этом кроме преобразования адреса инициатора соединения будет преобразоваться и порт, с которого устанавливается соединение в любой свободный порт, указанный в диапазоне port1:port2 опции portmap (в примере 10000-20000). Это исключает ситуацию, когда при попытке нескольких адресов в сети *net1* установить соединение с внешним ресурсом через МЭ, используя одинаковый

[1]На сегодняшний момент зарезервированным адресным пространством, предназначенным для использования вне сети Интернет, являются следующие адреса:

- 10.0.0.0 — 10.255.255.255
- 172.16.0.0 — 172.16.255.255
- 192.168.0.0 — 192.168.255.255

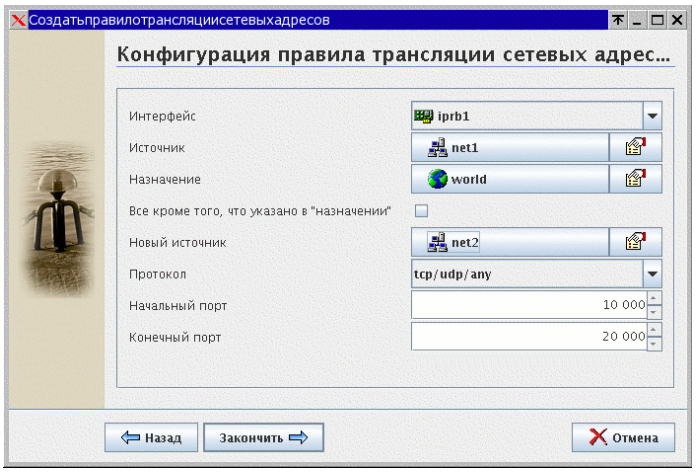

Рис. 5.3: Правила динамической трансляции адресов

порт отправителя при инициализации соединения, соединение устанавливается только для первого адреса, успевшего раньше других зарезервировать этот порт под свои нужды.

Учитывая, что некоторые протоколы, например, ICMP или IPSec, не могут использовать этот механизм из-за отсутствия портов в этих протоколах, в качестве значения в поле `Протокол` установлен `tcp/udp/any`.

5.2.2 Статическая сетевая трансляция адресов

Статическая сетевая трансляция адресов, или `Static NAT`, предназначена для двусторонней трансляции адресов по схеме «один к одному». Данный режим применяется для организации равноправного взаимодействия с сетями, защищенными межсетевым экраном. Под равноправным взаимодействием подразумевается то, что инициировать сессию может компьютер как из транслируемой сети, так и любой компьютер из внешней сети. Этим режим статической трансляции адресов отличается от режима `Динамической сетевой трансляции адресов`, при применении которого инициировать сессию может компьютер только из защищаемой сети.

Правило статической трансляция адресов имеет следующий вид:

```
bimap <Интерфейс> <Источник> -> <Назначение>
```

где:

<Интерфейс>
: Имя интерфейса, на котором будет выполняться трансляция адресов. Статическая трансляция должна производиться на исходящем интерфейсе.

<Источник>, <Назначение>
: Адреса сетей или индивидуальные IP-адреса, между которыми устанавливается соответствие.

Как видно из записи правила, адреса из сети *Источник* при прохождении через сетевой интерфейс МЭ заменяются на соответствующие адреса из сети *Назначение*. При обращении к адресам из сети *Назначение* межсетевой экран переадресует их к соответствующим адресам из сети *Источник*, то есть между адресами из сети *Источник* и *Назначение* устанавливается взаимооднозначное соответствие. Следует отметить, что размер сетей в правиле статической трансляции должен быть одинаков. [1]

Правила статической трансляции адресов, как правило, применяются для обеспечения взаимодействия сетей без изменения их внутренней адресации. Предположим, существует организация, использующая сеть 10.0.0.0/8, а ее филиал, до этого не объединенный с головной организацией в единую сеть, использует сеть *demonet* (192.168.1.0/24). При организации общей сети можно объединить филиал с головной организацией без изменения адресного пространства филиала или маршрутизации, используя статическую трансляцию адресов. Для этого необходимо установить соответствие между свободной подсетью в сети 10.0.0.0 (*mainnet*, адрес 10.10.10.0/24) и сетью филиала 192.168.10.0/24 (рисунок 5.4).

Не следует применять статическую трансляцию для обеспечения доступа к серверам внутренней сети из сети Интернет, так как такое правило автоматически обеспечивает не только доступ к этим серверам из внешних сетей, но и возможность доступа от серверов во внешние сети, что, как правило, совершенно не нужно. Для решения этих проблем правильнее применять правила трансляции адресов типа «Переадресация — прозрачный шлюз».

[1] При статической трансляции адресов необходимо обеспечить маршрутизацию этих адресов через МЭ. Варианты решения данной задачи описаны в разделе 5.2.5 на стр. 59

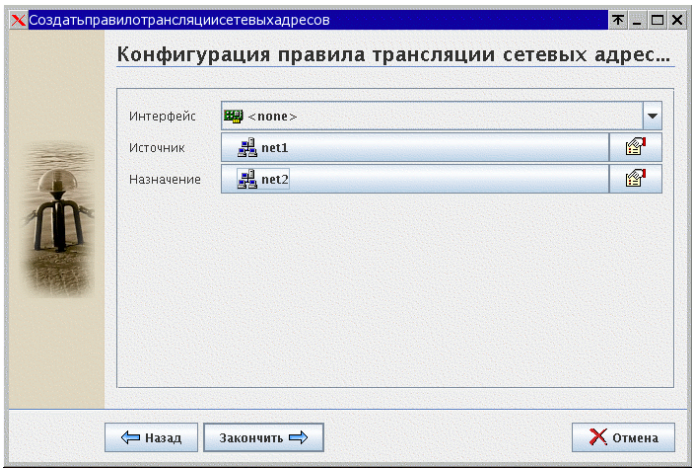

Рис. 5.4: Правила статической трансляции адресов

5.2.3 Переадресация — прозрачный шлюз

Правила трансляции адресов данного типа, в отличие от предыдущих, изменяют не адрес и/или порт инициатора соединения, а адрес и/или порт получателя соединения. Правила такого типа используются для обеспечения «прозрачного» режима работы прикладных шлюзов или для организации доступа к серверам, находящимся в защищаемой сети.

Правило переадресации имеет следующий вид:

```
rdr <interface> <network1> port <port1> -> <address> port <port2>
```

где:

<Интерфейс>
Имя интерфейса, на котором будет выполняться трансляция адресов. Переадресация должна производиться на входящем интерфейсе.

<Источник>, <Порт назначения>
Адрес сети и номер порта, при установлении соединения к которым будет происходить трансляция адресов.

<Новое назначение>, <Новый порт назначения>
На этот адрес и порт будет перенаправлено устанавливаемое соединение.

Наиболее часто правила данного типа используются для организации доступа ко внутренним серверам из внешних сетей. Предположим, что у нас во внутренней сети установлен HTTP-сервер *microball* с адресом 192.168.11.177, и нам необходимо предоставить к нему доступ из сети Интернет. Для этого мы создаем правило, переадресующее все соединения к внешнему интерфейсу межсетевого экрана, порт 80, на наш внутренний HTTP-сервер (рисунок 5.5).

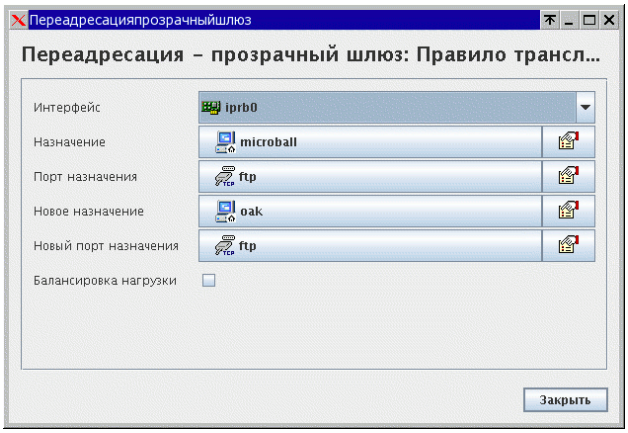

Рис. 5.5: Правила переадресации пакетов

Также правила данного типа применяются для обеспечения «прозрачного» режима работы прикладных шлюзов. Происходит это следующим образом:

- пакетный фильтр перенаправляет пакеты, направляющиеся к серверу, на интерфейс МЭ, на котором принимает соединения прикладной шлюз;
- прикладной шлюз обрабатывает запрос;

- перед тем, как отправить запрос, прикладной шлюз «спрашивает» у подсистемы трансляции адресов куда первоначально отправлялся пакет;
- прикладной шлюз отправляет запрос по полученному адресу.

Предположим, что мы хотим обеспечить «прозрачный» режим работы прикладного шлюза *ftp-gw*, принимающего соединения на внутреннем интерфейсе iprb1 межсетевого экрана. Для этого мы должны создать следующее правило (рисунок 5.6):

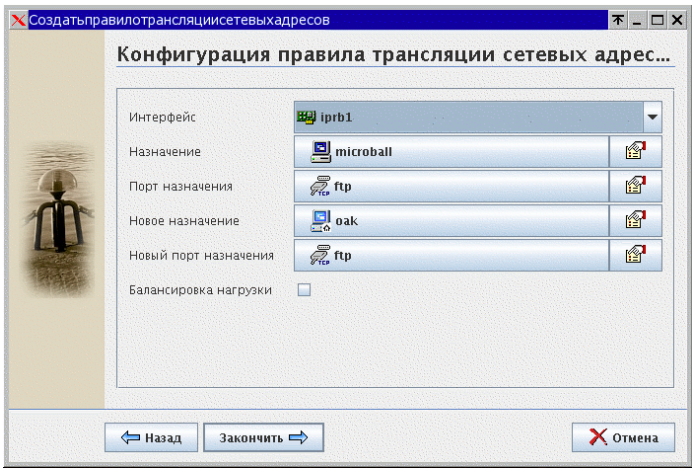

Рис. 5.6: Правило для «прозрачного» режима работы прикладного шлюза протокола FTP

Создавая правила переадресации, следует помнить, что нельзя пользоваться такими правилами как «отражателем». Например, такое правило работать не будет:

```
rdr iprb1 192.168.10.10/32 port 80 -> 192.168.10.6 port 80
```

так как и 192.168.10.10 и 192.168.10.6 находятся в одном сетевом сегменте и система не сможет произвести маршрутизацию подобного трафика.

Также не следует в качестве нового адреса назначения указывать `localhost` (127.0.0.1). Из-за особенностей реализации IP-стека в ядре ОС `Solaris` пакетный фильтр не может выполнить переадресацию пакетов на локальный интерфейс (`loopback`).

5.2.4 Шлюз уровня ядра

Существует ряд протоколов прикладного уровня, работа с которыми только средствами универсального IP-фильтра невозможна. Как правило, это протоколы, передающие информацию транспортного характера на прикладном уровне. Например, FTP-клиент передает серверу информацию о том, на каком порту он будет принимать данные с помощью команды *PORT*, параметрами которой являются IP-адрес и номер порта. Пакетный фильтр, для того чтобы обеспечить работу по протоколу FTP, должен знать об этой особенности протокола, и «увидев» команду *PORT*, открыть доступ к клиенту по указанным в этой команде параметрам.

Для решения проблем подобного характера в пакетном фильтре существуют шлюзы приложений ядерного уровня, или kernel proxy, обеспечивающие обмен и фильтрацию данных для протоколов, подобных FTP. Шлюзы приложений ядерного уровня существуют для протоколов:

- FTP
- Berkeley r-services (rlogin, rsh, rexec)
- Real Audio (только протокол PNA)
- IPSec
- NETBIOS
- H.323

Следует заметить, что шлюзы приложений ядерного уровня не являются полноценными прикладными шлюзами, так как не обеспечивают проверку протокола обмена. Их основная задача — позволить работу по данному протоколу через межсетевой экран. С другой стороны, функционируя на уровне пакетного фильтра, шлюзы приложений ядерного уровня обеспечивают более высокую скорость обмена данными, чем классические прикладные шлюзы, поэтому использовать их следует в тех случаях, когда важна в первую очередь скорость обмена через межсетевой экран. Во всех остальных случаях предпочтительнее использовать классические прикладные шлюзы.

Правило `Шлюза уровня ядра` имеет следующий вид:

```
map <Интерфейс> <Источник> -> <Назначение> <Новый источник> proxy port
<Порт> <Шлюз уровня ядра>
```

где:

<Интерфейс>
Имя интерфейса, на котором будет выполняться трансляция адресов. Трансляция адресов данного типа должна производиться на внешнем интерфейсе.

<Источник>
Адрес сети, для которой будет производиться трансляция адресов

<Назначение>
Адрес сети или хоста, куда направляется пакет.

<Новый источник>
Адрес, в который будут транслироваться адреса из сети, указанной в пункте `Источник`.

<Порт>
Порт, на котором устанавливается соединение.

<Шлюз уровня ядра>
Тип используемого прикладного шлюза:
- ftp/tcp
- rcmd/tcp
- raudio/tcp
- netbios/tcp
- h323/tcp
- ipsec/udp

Предположим, мы хотим обеспечить работу нашей внутренней сети *net1* (172.16.0.0/16) по протоколу FTP таким образом, чтобы все адреса из этой резервной сети преобразовывались в адрес внешнего интерфейса iprb0 *playball-iprb0*. Для этого необходимо создать правило `Шлюз уровня ядра` следующего вида (рисунок 5.7):

Следует отметить, что это правило должно быть помещено до остальных правил NAT.

5.2.5 Трансляция адресов и маршрутизация

Чтобы трансляция адресов работала корректно, вы должны убедиться, что ответы на те пакеты, чей адрес был транслирован, возвращаются к отправителю (исключение составляет вариант, когда трансляция адресов производится в IP адрес внешнего интерфейса МЭ). Предположим, что у нас имеется внутренняя сеть 192.168.10.0/24, и все адреса из этой сети мы динамически транслируем в

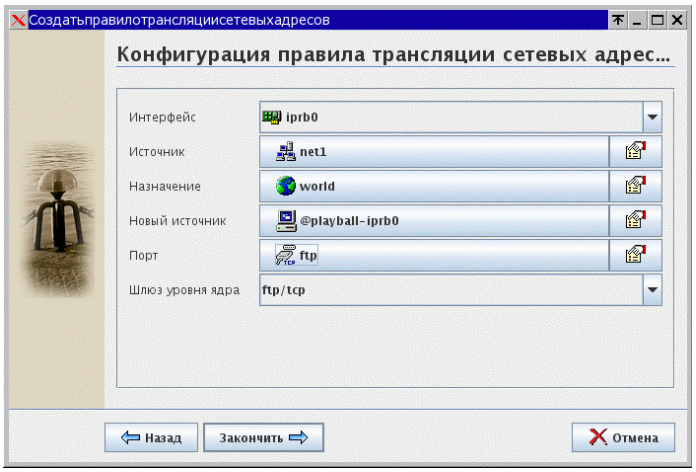

Рис. 5.7: Правила Шлюза уровня ядра

пул адресов 194.87.88.35 — 194.87.88.37 на внешнем интерфейсе МЭ, имеющим адрес 194.87.88.34. Проблема заключается в том, что маршрутизатор за межсетевым экраном должен знать, куда отправлять пакеты, чей адрес назначения является транслированный адрес. Существует два варианта решения этой проблемы.

1. Необходимо изменить правила на маршрутизаторе. Например, в предыдущем примере необходимо создать правило маршрутизации, переадресующее все пакеты, чей адрес назначения является адреса 194.87.88.35 — 194.87.88.37 на интерфейс межсетевого экрана, на котором производится статическая трансляция.
2. Изменить ARP-таблицу на межсетевом экране. Этот метод заключается в том, чтобы МЭ анонсировал свой MAC-адрес для транслированных адресов. Этот метод больше подходит для небольших блоков адресов. В этом случае на МЭ необходимо добавить следующие записи в ARP таблицу:

```
arp  -s 194.87.88.35 <Firewall MAC address> pub
arp  -s 194.87.88.36 <Firewall MAC address> pub
arp  -s 194.87.88.37 <Firewall MAC address> pub
```

Firewall MAC address в этом примере — MAC-адрес интерфейса, на котором производится трансляция адресов, в нашем случае — 194.87.88.34.

5.3 Программы управления пакетным фильтром

Программы управления пакетным фильтром позволяют осуществлять управление пакетным фильтром из командной строки. С помощью этих команд можно менять конфигурацию правил пакетного фильтра, конфигурацию правил подсистемы трансляции адресов, осуществлять сбор статистики и диагностику работы пакетного фильтра.

5.3.1 Программа ipf

Программа ipf предназначена для загрузки и изменения набора правил фильтрации в пакетном фильтре.

5.3.1.1 Принципы работы

ipf представляет собой утилиту командной строки, позволяющую управлять пакетным фильтром. При штатной работе ipf не используется, поскольку все управление осуществляется через графический интерфейс.

Программа ipf открывает файлы, указанные с помощью опции -f (при этом знак «-» трактуется как стандартный ввод программы) и производит синтаксический разбор перечисленных там правил. Если правило не содержит синтаксических ошибок, данное правило добавляется или удаляется во внутренний список ядра. Если правило содержит ошибки, то оно игнорируется, на стандартный вывод выводится сообщение об ошибке, после чего обрабатывается следующее правило. Добавление или удаление правил во внутренний список ядра производится в том порядке, в котором они указаны в обрабатываемом файле.

5.3.1.2 Синтаксис вызова программы ipf

Общий синтаксис вызова программы ipf выглядит следующим образом:

```
/usr/sbin/ipf [ -6AdDEInrsUvVyzZ ] [ -l <block|pass|nomatch> ]
[ -F <i|o|a|s|S> ] -f <имя файла> [ -f <имя файла> [...]]
```

5.3.1.3 Список опций программы ipf

-6 Требуется для обработки и последующей загрузке правил протокола IPv6;

-A Изменять текущий (активный) набор правил. Данная опция подразумевается по умолчанию;

-d Включить отладочную информацию. При включенной отладочной информации загружаемые правила отображаются на стандартном выводе в шестнадцатеричном виде;

-F <i|o|a> Определяет, какое подмножество правил фильтрации необходимо обнулить. Параметром может быть i (input), o (output) или a (all, обнулить все правила);

-F <s|S> Обнуляет записи в таблице состояния виртуальных соединений. При использовании параметра s обнуляются записи только о не полностью установленных виртуальных соединениях, параметр S очищает таблицу состояния виртуальных соединений полностью;

-f <имя файла> Загрузить или удалить список правил из указанного файла;

-l <pass|block|nomatch> Включает протоколирование пакетов, обрабатываемых модулем ipfilter. При использовании опции pass протоколируется пакеты, пропущенные модулем, block — заблокированные модулем. При использовании опции nomatch протоколируются пакеты, параметры которых не соответствуют ни одному правилу фильтрации, загруженному в модуль ipfilter;

-n Не загружать указанный набор правил, произвести только синтаксический разбор;
-r Удалить заданный список правил;
-s Поменять текущий и резервный набор правил;
-U Заблокировать пакеты, не принадлежащие стеку протоколов IP. Сообщения о таких пакетах выводится на системную консоль;
-v Включить режим подробного отображения информации (verbose mode);
-V Показать номер версии модуля ipfilter;
-y Принудительно синхронизировать ядерный список интерфейсов ОС со внутренним списком интерфейсов модуля ipfilter;
-z Для каждого правила в указанном списке показать накопленную статистику на этот момент и обнулить статистическую информацию;
-Z Обнулить всю накопленную статистику для правил фильтрации (статистика по обработанным фрагментам и таблице виртуальных состояний не обнуляется).

5.3.1.4 Примеры использования программы ipf

Сбросить все текущие правила фильтрации и загрузить новые из файла *./etc/opt/fw/ipf.conf*:

```
ipf -Fa -f /etc/opt/fw/ipf.conf
```

Следует заметить, что при этом уже установленные соединения не сбрасываются.

Сбросить правила фильтрации и все установленные соединения:

```
ipf -Fa -Fs
```

Сбросить правила фильтрации, указанные в файле *./etc/opt/fw/old.conf*:

```
ipf -r -f /etc/opt/fw/old.conf
```

Вывести на экран версию пакетного фильтра:

```
ipf -V

ipf: IP Filter: v4.1.8 (592)
Kernel: IP Filter: v4.1.8
Running: yes
Log Flags: 0 = none set
Default: pass all, Logging: available
Active list: 1
Feature mask: 0x187
```

5.3.2 Программа ipnat

Программа ipnat предназначена для загрузки и изменения набора правил трансляции адресов (NAT) в пакетном фильтре.

5.3.2.1 Принципы работы

`ipnat` представляет собой утилиту командной строки, позволяющую управлять правилами трансляции адресов в пакетном фильтре. При штатной работе `ipnat` не используется, поскольку все управление осуществляется через графический интерфейс.

Программа `ipnat` открывает файлы, указанные с помощью опции `-f` (при этом знак «-» трактуется как стандартный ввод программы), и производит синтаксический разбор перечисленных там правил. Если правило не содержит синтаксических ошибок, данное правило добавляется или удаляется во внутренний список ядра, содержащий правила трансляции адресов. Если правило содержит ошибки, то оно игнорируется, на стандартный вывод выводится сообщение об ошибке, после чего обрабатывается следующее правило. Добавление или удаление правил во внутренний список ядра производится в том порядке, в котором они указаны в обрабатываемом файле.

5.3.2.2 Синтаксис вызова программы `ipnat`

Общий синтаксис вызова программы `ipnat` выглядит следующим образом:

```
/usr/sbin/ipnat [ -lnrsvCF ] -f <имя файла>
```

5.3.2.3 Список опций программы `ipnat`

-C Удалить все текущие правила трансляции адресов;
-F Удалить все созданные на текущий момент записи в таблице состояния трансляции адресов;
-l Показать содержимое таблицы состояния трансляции адресов;
-n Не загружать указанный набор правил, произвести только синтаксический разбор;
-s Отобразить статистику NAT;
-r Удалить заданный набор правил трансляции из текущего набора правил NAT;
-v Включить режим подробного отображения информации о работе системы трансляции адресов (verbose mode);

5.3.2.4 Примеры использования программы `ipnat`

Сбросить все текущие правила трансляции и загрузить новые из файла *`/etc/opt/fw/ipnat.conf`*:

```
ipnat -FC -f /etc/opt/fw/ipnat.conf
```

Получить статистику по работе подсистемы трансляции адресов:

```
ipnat -s

mapped  in      4410090 out     3683316
added   404280  expired 400488
no memory       0       bad nat 0
inuse   318
rules   7
wilds   0
```

Следует обратить внимание на поле `no memory` — если значение этого поля отлично от нуля, это значит, что для работы подсистемы трансляции адресов не хватает памяти.
Размер выделяемой памяти устанавливается параметром ipf_nattable_sz в файле */etc/system* :

```
* ipf: ajust NATTABLESIZE
set ipf:ipf_nattable_sz = 20021
```

Примечание После изменений параметров в файле */etc/system* систему необходимо перезагрузить.

5.3.3 Программа ipmon

Программа `ipmon` предназначена для получения протоколируемой информации от пакетного фильтра.

5.3.3.1 Принципы работы

Программа ipmon открывает для чтения устройство `/dev/ipl`, в которое поступают данные от модуля `ipfilter`. Данные, считанные с устройства `/dev/ipl` переводятся программой из внутреннего двоичного представления в текстовую форму, приемлемую для просмотра и обработки. В зависимости от опций, с которыми запущена программа `ipmon`, выходная информация может быть направлена на стандартный вывод, в файл или в подсистему протоколирования `syslog`. По умолчанию информация от `ipmon` направляется на стандартный вывод.

5.3.3.2 Список опций программы ipmon

-a Получать информацию от всех подсистем модуля `ipfilter` (подсистемы трансляции адресов, подсистемы сохранения виртуальных соединений, подсистемы IP фильтрации)

-D Запустить программу `ipmon` как отсоединенный процесс (`daemon`)

-f <имя устройства> Использовать указанное устройство в качестве источника информации от модуля `ipfilter`. По умолчанию используется устройство `/dev/ipl`

-F Очистить текущий буфер устройства `/dev/ipl` (или другого, указанного с помощью опции `-f`)

-n Отображать доменные адреса и имена сервисов в символьном виде, если это возможно

-N <имя устройства> Получать информацию только от подсистемы трансляции адресов (`NAT`) с указанного устройства

-o Указать, от какой подсистемы получать информацию для протоколирования. `N` - от подсистемы трансляции адресов, `S` - от подсистемы сохранения виртуальных соединений (`state system`), `I` - от подсистемы IP фильтрации. Опция `-a` эквивалентна опции `-o NSI`

-O Указать, от какой подсистемы не получать информацию для протоколирования. Именование подсистем аналогичны используемым в опции `-o`

-p При использовании этой опции номер порта в файлах журналов всегда будет представлен в виде числа и не будет предпринято попыток поиска его символьного обозначения в файле `/etc/services`

-P Записать идентификатор процесса `ipmon` в указанный файл. По умолчанию идентификатор процесса записывается в файл `/etc/opt/fw/ipmon.pid`.

-s Информация от программы `ipmon` будет перенаправлена в подсистему протоколирования `syslog`. Информация выводится с типом сообщения (`facility`) `local0` и приоритетами `LOG_INFO`, `LOG_NOTICE`, `LOG_WARNING` и `LOG_ERR`.

-S <имя устройства> Получать информацию только от подсистемы сохранения виртуальных соединений (`state system`) с указанного устройства.

-t Считывать информацию из устройства подобно программе `tail(1)`.

-v Отображать размер `TCP`-окна и содержимого полей `ACK` и `SEQUENCE` (только для протокола `TCP`).

-x Отображать содержимое передаваемых пакетов в шестнадцатеричном виде.
-X Отображать содержимое заголовков пакетов в шестнадцатеричном виде.

5.3.3.3 Запуск программы ipmon

Программа ipmon запускается автоматически при загрузке системы с опцией -s, таким образом вся информация от модуля ipfilter поступает в подсистему протоколирования syslog.

5.3.4 Программа ipfstat

Программа ipfstat предназначена для получения статистики работы пакетного фильтра, а также списков текущих правил фильтрации.

5.3.4.1 Принципы работы

Программа ipfstat использует устройство /dev/kmem для получения статистики работы пакетного фильтра. Для корректной работы программе ipfstat необходим доступ на чтение как /dev/kmem, так и непосредственно ядра ОС (по умолчанию используется /vmunix.

При использовании опции «-t» вывод программы ipfstat подобен программе *top*. В этом режиме можно использовать опции «-C, -D, -P, -S, -T» для фильтрации выводимой информации (более подробно эти опции описаны в разделе 5.3.4.2 на стр. 66).

Примечание Для корректной работы программы ipfstat в этом режиме ширина экрана должна быть как минимум 80 символов.

При работе программы ipfstat с опцией «-t» можно использовать следующие команды:

d Выбор типа отображаемой информации
l Перерисовать экран
q Выйти из программы
s Переключение между режимами сортировки (по протоколам, по пакетам, по количеству переданных байт и т.д.)
r Включить/выключить реверсный режим сортировки

5.3.4.2 Синтаксис вызова программы ipfstat

Общий синтаксис вызова программы ipfstat выглядит следующим образом:

```
/usr/sbin/ipfstat [ -6aAfghIinosv ] [ -d <устройство> ]

/usr/sbin/ipfstat -t [ -C ] [ -D <адрес и порт> ] [ -P <протокол> ]
[  -S <адрес и порт> ] [ -T <время обновления> ] [ -d <устройство> ]
```

5.3.4.3 Список опций программы ipfstat

-6 Показать список правил для протокола IPv6 (если они есть).
-a Отобразить информацию по учету трафика. Данная опция показывает количество переданных байт для каждого правила типа «count».

-A Отобразить аутентификационную статистику (в данный момент не используется, зарезервировано на будущее).

-C Используется только совместно с опцией «-t». Отображает «закрытые» виртуальные TCP соединения. В обычном режиме TCP соединения, достигшие состояния *CLOSE_WATI* не отображаются. Опция «-C» позволяет отображать такие соединения наравне с остальными.

-d <имя устройства> Использовать для работы устройство <имя устройства>. По умолчанию используется устройство /dev/ipl.

-D <адрес и порт> Используется только совместно с опцией «-t». Показывать только те виртуальные соединения, чей адрес и порт назначения соответствуют указанным. Адрес и порт указываются в виде *ip_address[,port]* и могут быть либо в цифровом виде либо как «any», где «any» подразумевает любой адрес или порт соответственно. По умолчанию используется «-D any,any»

-f Показать информацию по фрагментированым пакетам.

-g Показать сконфигурированные группы правил пакетного фильтра, как активные, так и неактивные.

-h Показать таблицу правил пакетного фильтра с указанием, сколько раз каждое правило было использовано. Используется в комбинации с опциями «-i» или «-o».

-i Отобразить таблицу правил пакетного фильтра для входящих соединений.

-I Переключает отображение активных и неактивных наборов правил фильтрации.

-n Показывать номер правила при его выводе.

-o Отобразить таблицу правил пакетного фильтра для исходящих соединений.

-P <протокол> Используется только совместно с опцией «-t». Показывать только те виртуальные соединения, которые используют указанный протокол. Аргументом этой опции может быть имя протокола (определенное в файле /etc/protocols) или его номер. Если эта опция не указана, отображаются виртуальные соединения для всех протоколов.

-s Отобразить статистику по работе подсистемы «сохранения состояний».

-S <адрес и порт> Используется только совместно с опцией «-t». Показывать только те виртуальные соединения, чей адрес и порт инициатора соединения соответствуют указанным. Адрес и порт указываются в виде *ip_address[,port]* и могут быть либо в цифровом виде либо как «any», где «any» подразумевает любой адрес или порт соответственно. По умолчанию используется «-S any,any».

-t Показывать таблицу состояния виртуальных соединений подобно тому, как программа *top* отображает список процессов.

-T <время обновления> Используется только совместно с опцией «-t». Указывает время (в секундах), через которое информация, выводимая на экран программой *ipfstat* должна обновляться.По умолчанию используется «-T 1».

-v Отображать отладочную информацию

5.3.4.4 Примеры использования программы ipfstat

Показать текущие правила пакетного фильтра для входящих соединений:

```
ipfstat -hi

7 pass in quick proto tcp from 10.5.5.0/24 to 10.5.1.0/24
flags S/FSRPAU keep state keep frags
0 pass in quick proto udp from 10.5.5.0/24 to 10.5.1.0/24
keep state keep frags
3 pass in quick proto icmp from 10.5.5.0/24 to 10.5.1.0/24
keep state keep frags
0 pass in quick on hme1 proto icmp from 192.168.193.0/24
to 192.168.192.10/32 keep state keep frags
208 pass in quick on hme1 proto tcp from 192.168.193.0/24
```

```
to 192.168.192.10/32 flags S/FSRPAU keep state keep frags
364 pass in quick on hme1 proto udp from 192.168.193.0/24
to 192.168.192.10/32 keep state keep frags
0 pass in quick proto tcp from 195.5.134.46/32 to 217.151.129.34/32
port = 8070 flags S/FSRPAU keep state keep frags
93173 pass in quick proto tcp from 10.5.1.4/32
to any flags S/FSRPAU keep state keep frags
46386 block in log first level notice quick from any to any
```

Число перед началом правила указывает, сколько раз использовалось данное правило

Показать таблицу состояния виртуальных соединений

```
ipfstat -t

                 relay - IP Filter: v3.4.29 - state top
Src = 0.0.0.0  Dest = 0.0.0.0  Proto = any  Sorted by = # bytes

Source IP          Destination IP      ST   PR  #pkts   #bytes  ttl
10.5.1.96,53       10.5.1.4,3300       0/0  udp 6807    707233  1:28
10.5.5.17,1104     10.5.1.50,139       4/4  tcp 782     136106  25:48:28
10.5.1.24,2468     10.5.1.96,8888      4/4  tcp 134     86145   116:33:37
10.5.1.4,4696      205.188.8.254,443   4/4  tcp 526     69784   119:59:56
10.5.1.1,1861      10.5.1.96,23        4/4  tcp 1370    66702   119:20:47
10.5.1.142,2550    10.5.1.96,8888      4/4  tcp 829     59051   28:47:45
```

Получить статистику по работе подсистемы фильтрации :

```
ipfstat -s

IP states added:
        5499625 TCP
        333118 UDP
        19969 ICMP
        223728287 hits
        8481327 misses
        0 maximum
        0 no memory
        1547 bkts in use
        1849 active
        353082 expired
        5497781 closed
```

Следует обратить внимание на поля `maximum` и `no memory`. Число в поле `maximum` — сколько раз новая сессия не была добавлена из-за переполнения таблицы состояний, число в поле `no memory` — сколько раз новая сессия не была добавлена из-за нехватки памяти в системе. Если значение поля `maximum` отличается от нуля, необходимо увеличить размер таблицы состояний, как это описано в разделе Б на стр. 171.

Глава 6

Прикладные шлюзы

МЭ «Z-2» включает шлюзы прикладного уровня для протоколов HTTP, FTP, SMTP, POP3, TELNET, SNMP и Net8. МЭ также включает в себя прикладные шлюзы общего назначения, которые являются нейтральными по отношению к содержимому протокола и могут быть использованы для различных типов приложений, использующих в качестве транспорта протоколы TCP и UDP. `Универсальный прикладной шлюз TCP` и `Универсальный прикладной шлюз UDP` обеспечивают фильтрацию по сетевым адресам и портам источника и получателя запроса и протоколирование соединений. Шлюзы приложений могут также производить аутентификацию запроса на установление соединения на сервере аутентификации и авторизации. Разграничение доступа к шлюзам приложений производится с помощью списков управления доступом (Access Control Lists, ACL) на основании заданного диапазона IP-адресов и портов разрешенных источников запросов.

Все входящие в состав МЭ «Z-2» прикладные шлюзы используют общую схему работы со списками доступа (`Редактор ACL`). Данный раздел описывает общие принципы организации списков доступа и основные режимы функционирования прикладных шлюзов.

6.1 Функционирование резидентного процесса прикладного шлюза

Все прикладные шлюзы в межсетевом экране работают как резидентные процессы. Функционально схемы работы этих процессов делятся на три группы:

- Forked схема
- Preforked схема
- Udp схема

6.1.1 Forked схема

Принцип работы Forked схемы состоит в следующем:

- После запуска в системе функционирует один процесс прикладного шлюза, ожидающий соединения
- При поступлении соединения создается новый прикладной процесс *обработчик*, обслуживающий соединение.

Один обработчик обслуживает одно соединение и завершается при его закрытии. По такой схеме работают прикладные шлюзы протоколов SMTP, FTP и TELNET. В настройках шлюза указывается максимальное количество процессов обработчиков. В случае его превышения процессы обработчики создаваться не будут и соединения будут закрываться непосредственно после открытия.

Настройки резидентного процесса состоят из следующих пунктов:

Максимальное количество процессов-обработчиков — максимальное число процессов-обработчиков, обслуживающих соединения

Имя владельца процесса --- UID процесса.

Название группы процесса — GID процесса.

Предельное время ожидания ответа на запрос (в секундах) — максимальное время ожидания данных на соединении. По его истечении соединение будет закрыто.

Порт для входящих соединений — порт, на котором будет ожидаться запрос на установление соединения.

Интерфейс, на котором будут приниматься соединения — IP-адрес интерфейса, на котором будет ожидаться запрос на установления соединения. Если в качестве IP-адреса указано <any>, то запросы будут принимаются на всех интерфейсах межсетевого экрана.

Имя файла с новостью дня — полный путь к файлу, где указывается текст, выводимый после установления соединения.

Режим журналирования — режим, в котором будет производиться запись логов в журнал МЭ: INFO — информационные сообщения, DEBUG — сообщения, содержащие отладочную информацию WARNING — предупреждения.

6.1.2 Preforked схема

Принцип работы Preforked схемы отличается от описанной ранее Forked схемы. При старте запускается один *процесс-диспетчер* и группа *процессов-обработчиков*. Все свободные процессы-обработчики ожидают соединений пользователей. При поступлении соединения один из свободных обработчиков получает и обслуживает его. Каждый процесс-обработчик одновременно может обрабатывать одно соединение клиента. По окончании соединения процесс освобождается и продолжает ожидать поступления соединений. Процесс-диспетчер осуществляет функцию регулирования количества обработчиков. В настройках прикладного шлюза указывается начальное количество процессов обработчиков, максимальное их количество, рекомендуемое количество процессов обработчиков. При

поступлении соединения сверх установленного лимита, соединение встанет в очередь и будет обслужено первым свободным обработчиком, либо завершится по тайм-ауту.

В случае нехватки процессов-обработчиков диспетчер автоматически запускает новые, уведомляя об этом в системном журнале. Каждый запуск нового процесса занимает определенное время, таким образом, для достижения максимальной производительности и минимальных задержек нужно избегать старта новых процессов.

Данная схема работы минимизирует системные ресурсы, затрачиваемые на создание новых процессов обработчиков, тем самым значительно снижает нагрузку на систему при работе с протоколами, порождающими множество коротких сессий. По такой схеме работает прикладной шлюз протокола HTTP, универсальный прикладной шлюз протокола TCP, а также сервер аутентификации.

Настройки резидентного процесса состоят из следующих пунктов:

Число процессов-обработчиков, запускаемых при их нехватке — число процессов-обработчиков запросов, запускаемых в случае нехватки обработчиков.

Требуемое количество процессов-обработчиков — требуемое администратором число процессов-обработчиков.

Максимальное количество процессов-обработчиков — максимальное число процессов-обработчиков. Таким способом задается верхняя граница количества одновременно обслуживаемых запросов.

Начальное количество процессов-обработчиков — количество процессов, запускаемых при старте.

Имя владельца процесса — UID процесса, от которого будут обслуживаться запросы

Название группы процесса — GID процесса, от которого будут обслуживаться запросы

Предельное время ожидания ответа на запрос (в секундах) — максимальное время ожидания данных на соединениях. По его истечении соединение будет закрыто.

Порт для входящих соединений — порт, на котором будет ожидаться запрос на установление соединения.

Интерфейс, на котором будут приниматься соединения — IP-адрес интерфейса, на котором будет ожидаться запрос на установления соединения. Если в качестве IP-адреса указано <any>, то запросы будут принимаются на всех интерфейсах межсетевого экрана.

Максимальное количество запросов на процесс (0 - неограничено) — максимальное количество соединений, которое может обработать один процесс-обработчик за все время своей работы. Ноль означает неограниченное количество. По истечение этого лимита процесс завершается и при необходимости диспетчером будет запущен новый процесс. Эта опция применяется в сервере аутентификации при работе с PAM модулями, имеющими утечки памяти.

Режим журналирования — режим, в котором будет производиться запись логов в журнал МЭ: INFO — информационные сообщения, DEBUG — сообщения, содержащие отладочную информацию WARNING — предупреждения.

6.1.3 UDP схема

В случае использования UDP-схемы при обработке UDP-датаграмм не создается новых процессов. Вся пересылка пакетов производится в одном процессе. В настройках шлюза задается размер таблицы состояний псевдосессий

Максимальное количество состояний UDP — размер таблицы состояний. Задает максимальное количество одновременно обслуживаемых *псевдосессий* udp.

Имя владельца процесса — UID процесса, от которого будут обслуживаться запросы.

Название группы процесса — GID процесса, от которого будут обслуживаться запросы.

Предельное время ожидания ответа на запрос (в секундах) — максимальное время ожидания ответа в *псевдосессии*.

Порт для входящих соединений — порт, на котором будут ожидаться udp-датаграммы.

Интерфейс, на котором будут приниматься соединения — IP-адрес интерфейса, на котором будет ожидаться запрос на установления соединения. Если в качестве IP-адреса указано <any>, то запросы будут принимаются на всех интерфейсах межсетевого экрана.

Режим журналирования — режим, в котором будет производиться запись логов в журнал МЭ: INFO — информационные сообщения, DEBUG — сообщения, содержащие отладочную информацию, WARNING — предупреждения.

6.2 Списки доступа

Политика доступа состоят из набора *списков доступа* (*ACL*). Каждый список состоит из нескольких секций: списка разрешенных IP-адресов/портов входящих соединений (`Входящие правила`), списка разрешенных IP-адресов/портов исходящих соединений (`Исходящие правила`), дополнительных опций и других настроек, индивидуальных для каждого прикладного шлюза. Каждый список имеет свое уникальное в контексте шлюза имя. Обязательной частью списка доступа является раздел `Входящие правила`, присутствующий во всех шлюзах.

Механизм выбора списков доступа позволяет обслуживать разные категории клиентов в одном прикладном шлюзе.

6.2.1 Выбор списка доступа

Для каждого нового входящего на прикладной шлюз соединения выбирается список доступа на основе IP-адреса/порта клиента, инициирующего соединение, т.е. выбор происходит на основе представленной в разделе `Входящие правила` информации. Список для соединения выбирается один раз и остается неизменным в течении всего соединения (для шлюзов протокола `UDP` — на время обработки пакета).

Записи в секции `Исходящие правила` делятся на две группы — правила включения комбинации адреса/порт в ACL (`match`) и правила исключения комбинации адрес/порт из ACL (`exclude`).

Входящее соединение считается *попавшим* в ACL, если существует хотя бы одно правило включения, в которое попадает адрес и порт клиента, устанавливающего соединение, и не существует ни одного правила исключения, в которое попадает этот адрес и порт.

При обращении нового клиента поиск происходит в порядке записи списков сверху вниз и останавливается при первом удачном *попадании*. В случае если для клиента не найдено подходящего списка, соединения будет отвергнуто. Таким образом, все прикладные шлюзы используют политику *белого списка*.

6.2.2 Проверка исходящих соединений

Все прикладные шлюзы, кроме шлюза протокола `SMTP`, осуществляют исходящие соединения. В разделе `Исходящие правила` списка доступа содержатся правила проверки адреса/порта исходящих соединений.

Схема работы данного раздела в общем эквивалентна секции `Входящие правила`.

Записи делятся на две группы — разрешающие (*permit*) и запрещающие (*deny*).

Исходящее соединение будет разрешено, если IP-адрес и порт назначения попадает под хотя бы одну разрешающую запись и не попадает ни под одно запрещающее. В случае если комбинация IP-адрес и порт назначения не попадает ни под одну разрешающую запись, соединение не разрешается.

В шаблонах исходящих соединений задается протокол, по которому производится соединение. На данный момент только шлюз протокола *HTTP* поддерживает несколько протоколов исходящих соединений: http, ftp и ssl (раздел 6.6 на стр. 80).

6.2.3 Пример работы со списками доступа

Проиллюстрируем вышесказанное на примере. Предположим, необходимо обеспечить доступ из сети `internal` по протоколу FTP по следующей схеме:

1. Доступ с машины `Z-2 GUI` по FTP не ограничен;

2. Доступ из внутренней сети по FTP возможен только к корпоративному FTP-серверу `ftp.my-domain.ru`;
3. Доступ по FTP с машины `client` запрещен.

Для реализации данной политики доступа в прикладном шлюзе `ftp-gw` создадим два списка доступа — `admins` и `users`. В разделе `Входящие правила` списка доступа `admins` добавим машину `Z-2 GUI` (рисунок 6.1).

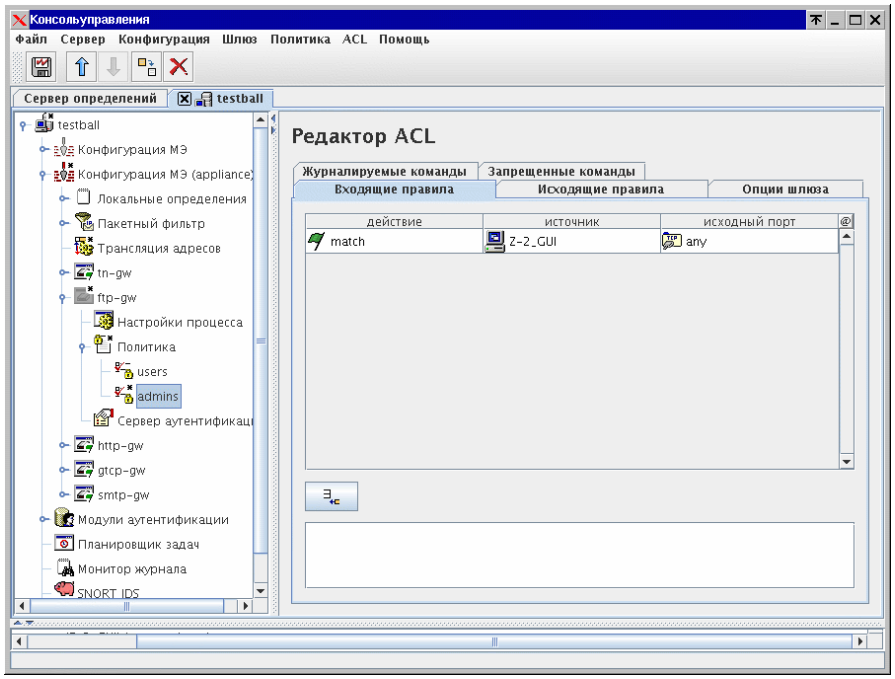

Рис. 6.1: Список доступа `admins`

В разделе `Исходящие правила` списка доступа `admins` разрешим доступ по протоколу FTP к любым адресам (рисунок 6.2).

В разделе `Входящие правила` списка доступа `users` разрешим доступ со всех машин из сети `internal`, кроме машины `client` (рисунок 6.3).

В разделе `Исходящие правила` списка доступа `users` разрешим доступ по протоколу FTP только к корпоративному FTP-серверу (рисунок 6.4).

Так как список доступа `admins` расположен до списка доступа `users`, все соединения сначала проверяются на принадлежность к этому списку доступа. Таким образом, хотя машина `Z-2 GUI` и входит в сеть `internal`, она попадет в список доступа `admins`, где нет ограничений на исходящие соединения. Все остальные машины из сети `internal` попадут в список доступа `users`, и смогут устанавливать FTP-соединения только с сервером `ftp.my-domain.ru`. Доступ к остальным адресам для них будет запрещен. Исключение составляет машина `client`, для которой установлено правило `exclude` в списке доступа. Таким образом, эта машина будет исключена из списка доступа `users`. Так как больше никаких списков доступа не определено, доступ по протоколу FTP для этой машины будет заблокирован.

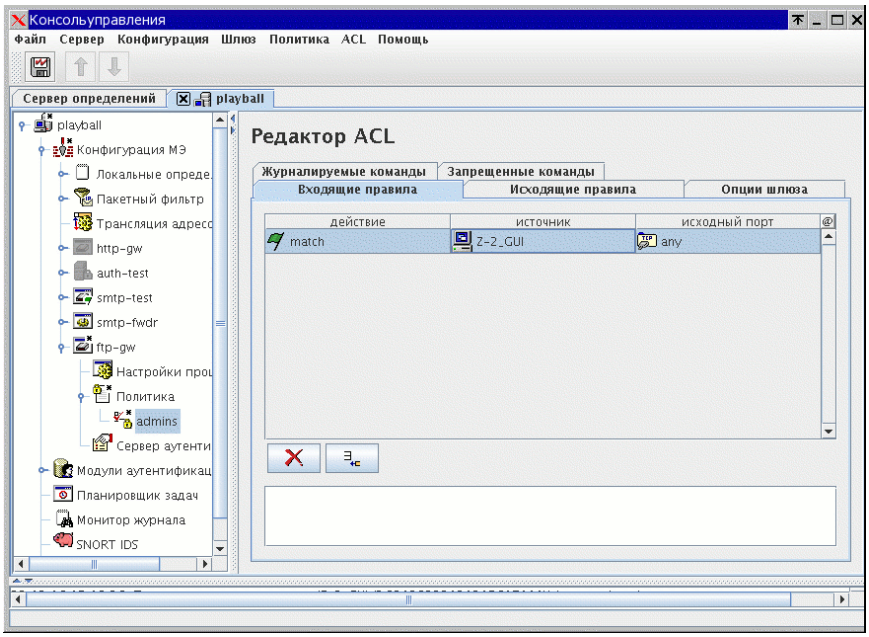

Рис. 6.2: Список доступа `admins`

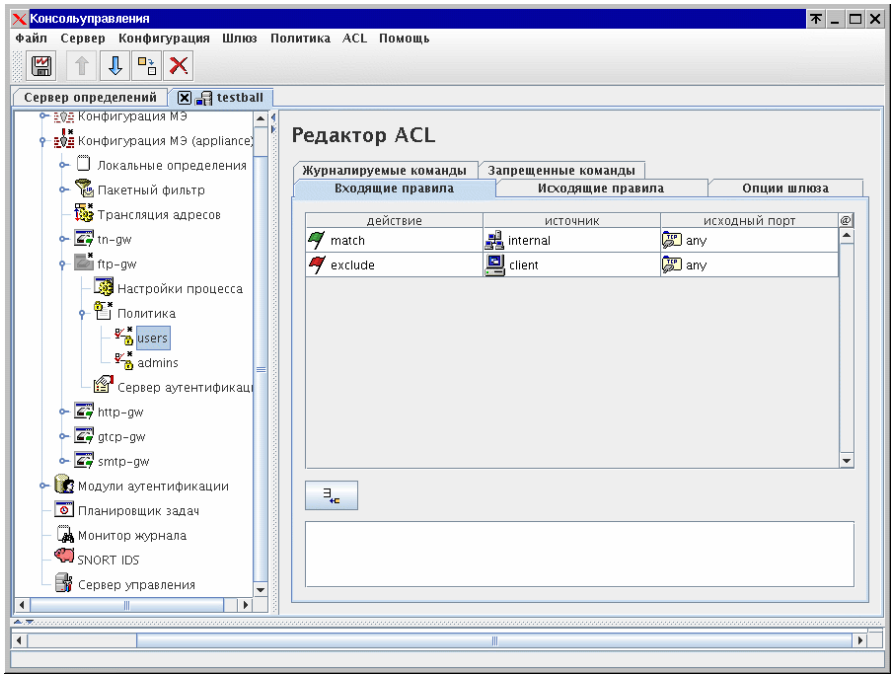

Рис. 6.3: Список доступа `users`

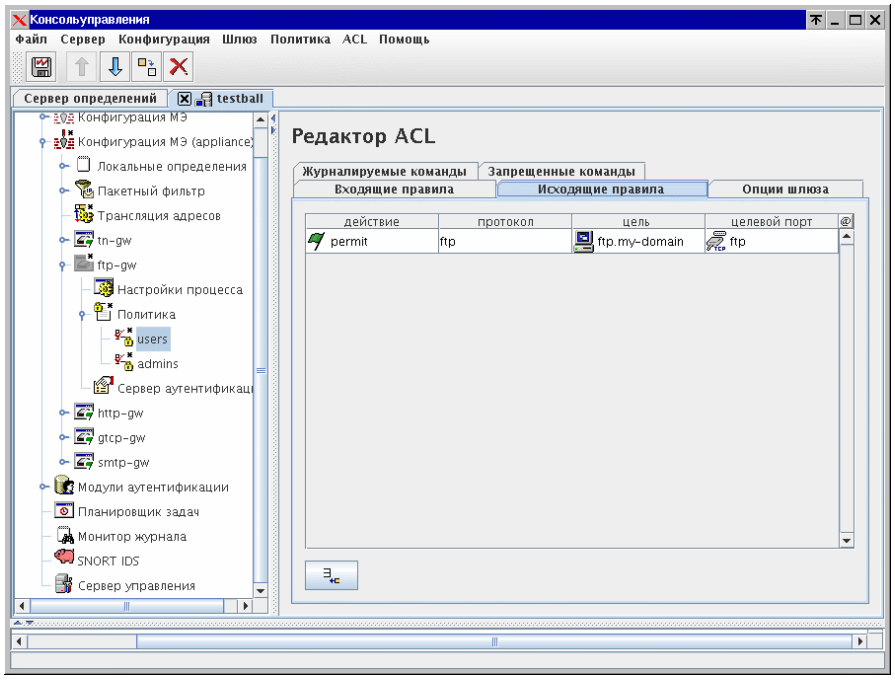

Рис. 6.4: Список доступа `users`

6.3 Функционирование прикладных шлюзов в «прозрачном» режиме

Каждый из шлюзов прикладного уровня может работать в одном из двух режимов — «прозрачном» и «непрозрачном». В «непрозрачном» режиме пользователь устанавливает соединение по соответствующему протоколу с МЭ Z-2 на заранее известный порт. На основании заданных правил фильтрации шлюз прикладного уровня разрешает или отвергает запрос на установление соединения. Если установление соединения данного пользователя разрешено, шлюз прикладного уровня позволяет пользователю перейти в режим соединения с удаленным сервером по соответствующему протоколу. В «прозрачном» режиме пользователь устанавливает соединение по соответствующему протоколу напрямую с требуемым сервером. Пакетный фильтр перехватывает запрос и передает его на обработку соответствующему шлюзу прикладного уровня, который на основании правил фильтрации разрешает или отвергает установление соединения. В этом режиме работа через МЭ полностью прозрачна для пользователя и не требует настройки клиентских мест или обучения персонала работе с межсетевым экраном.

Для обеспечения работы прикладных шлюзов в «прозрачном» режиме необходимо создать в настройках `NAT` пакетного фильтра правила трансляции адресов, перенаправляющие соединения от клиентов на прикладные шлюзы. Никаких дополнительных настроек для работы прикладных шлюзов в «прозрачном» режиме не требуется. Более подробно о настройках `NAT` для «прозрачного» режима можно узнать в разделе 5.2.3 на стр. 57.

6.4 Режим сохранения исходного адреса отправителя

При доступе к внутренним серверам организации через прикладные шлюзы все соединения, поступающие к этим серверам, имеют один и тот же адрес — адрес внутреннего интерфейса межсетевого экрана. Это может вызывать некоторые неудобства, так как лишает возможности администраторов этих серверов получать статистику, основанную на адресах клиентов, использующих этот сервер.

Для решения подобных проблем ряд прикладных шлюзов в МЭ «Z-2» могут функционировать в режиме сохранения исходного адреса отправителя (*force source address*). При работе в этом режиме прикладной шлюз устанавливает соединение с затребованным сервером не от своего имени, а от имени клиента, инициализирующего сессию. Это достигается путем динамического создания прикладным шлюзом правил трансляции сетевых адресов для пакетного фильтра. Таким образом, с одной стороны, обеспечивается доступ через прикладной шлюз, и с другой стороны сохраняется подлинность IP-адресов клиентов.

Режим сохранения исходного адреса отправителя на сегодняшний момент поддерживают следующие прикладные шлюзы:

- Универсальный прикладной шлюз TCP
- Прикладной шлюз telnet

Для активизации данного режима в настройках конфигурации резидентного процесса данных прикладных шлюзов необходимо в пункте «Устанавливать исходящий адрес клиента на интерфейсе» выбрать имя интерфейса, на котором будет происходить подмена IP-адреса межсетевого экрана на IP-адрес клиента.

Применять данный режим следует только при доступе внешних клиентов к внутренним серверам организации. Следует заметить, что при этом режиме необходима правильная настройка маршрутизации на серверах, к которым открывается доступ — маршрут к внешним машинам должен быть настроен через межсетевой экран. Если режим сохранения исходного адреса отправителя не используется, то делать этого как правило не обязательно, так как все соединения к защищаемым серверам инициируются с внутреннего IP-адреса межсетевого экрана.

6.5 Аутентификация пользователей

Прикладные шлюзы протоколов HTTP, TELNET, FTP и SMTP содержат возможность аутентификации пользователей.

Аутентификация осуществляется через специальный процесс — *сервер аутентификации*. Этот процесс обслуживает запросы на аутентификацию от прикладных шлюзов.

Сервер аутентификации не реализует непосредственно никаких схем аутентификации и не хранит паролей пользователей. Вместо этого используются PAM-модули — штатный механизм аутентификации ОС Solaris. PAM-модули, входящие в состав МЭ «Z-2», приведены в таблице 6.1:

Таблица 6.1: **PAM–модули**

Модуль	**Описание**
pam_db	аутентификация по схеме login-пароль с хранением базы пользователей в базе данных DBM
pam_skey	аутентификация одноразовыми паролями OTP. База данных пользователей хранится в текстовом файле
pam_radius	аутентификация пользователей на Radius сервере
pam_smb	аутентификация пользователей в домене Windows сети

В состав ОС Solaris также входит PAM-модуль pam_ldap для аутентификации на LDAP-сервере. Возможна также работа с PAM-модулями сторонних производителей, например, PAM_NDS от компании «Novell».

Прикладные шлюзы протоколов HTTP и SMTP поддерживают только аутентификацию по схеме login-пароль ввиду ограничений протокола.

Более подробная информация о сервере аутентификации и PAM-модулях приведена в разделе 7 на стр. 129.

6.6 Прикладной шлюз протокола HTTP

Прикладной шлюз протокола HTTP предназначен для осуществления и контроля обмена данными по протоколу HTTP. Он поддерживает протоколы HTTP 1.0 и 1.1 (последний — с поддержкой транспортных сессий).

Прикладной шлюз протокола HTTP осуществляет:

- Ограничение доступа по IP-адресу и порту сервера
- Ограничение доступа по URL
- Необходимую поддержку для блокировки баннеров
- Туннелирование протокола SSL с отдельной настройкой прав доступа по IP-адресу и порту
- Одностороннее ограниченное отображение FTP в HTTP
- Блокировку данных по MIME-типам
- Поддержку прозрачного режима
- Возможность трансляции запросов через «родительский» прикладной шлюз
- Блокирование заданных команд HTTP-протокола
- Проверку получаемых по FTP файлов на наличие вирусов
- Предупреждение администратора о посещении пользователем определенных URI
- Автоопределение типа некоторых файлов по их содержимому для блокировки определенных типов данных

6.6.1 Настройки прикладной шлюза протокола HTTP

Настройки прикладной шлюза протокола HTTP состоят из:

- настроек резидентного процесса (Конфигурация резидентного процесса), раздел 6.6.1.1 на стр. 80);
- настроек сервера аутентификации (Конфигурация сервера аутентификации, раздел 6.6.6 на стр. 85);
- настроек транспортных сессий (Конфигурация транспортных сессий, раздел 6.6.1.3.3 на стр. 83);
- настроек трансляции MIME-типов для FTP (раздел 6.6.7 на стр. 85);
- настроек правил доступа (раздел 6.6.1.2 на стр. 81).

Прикладной шлюз протокола HTTP осуществляет обработку трафика протокола HTTP в двух режимах — *прозрачном* и *непрозрачном*.

В *непрозрачном* режиме пользователь устанавливает соединение по протоколу HTTP с установленным на МЭ прикладным шлюзом HTTP. «Прозрачный» режим работы шлюза реализуется только совместно с пакетным фильтром.

6.6.1.1 Настройки резидентного процесса

Прикладной шлюз HTTP работает по preforked-схеме, т.е. одновременно в системе запущено множество процессов прикладного шлюза, слушающих и обслуживающих http-запросы. Настройки описаны в главе 6.1.2 на стр. 70.

Мы рекомендуем начинать подбор числа обработчиков с 20-30 процессов для сети средних размеров. Поскольку один клиент, как правило, открывает 2-3 соединения (например, так делают Internet Explorer и Netscape), число процессов должно быть не меньше 5-10.

6.6.1.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к прикладному шлюзу протокола `http` производится с помощью раздела `Политика`. Список доступа привязан к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Список доступа прикладного шлюза HTTP состоит из:

- раздела `Входящие правила`, где определяются правила контроля входящих соединений (раздел 6.2 на стр. 73);
- раздела `Исходящие правила`, где определяются правила контроля исходящих соединений;
- раздела `Опции шлюза`, где задаются опции правила:
  - `Включить авторизацию на шлюзе` включает аутентификацию пользователей на прикладном шлюзе (раздел 6.6.6 на стр. 85);
  - `Хост родительского прикладного шлюза` указывает родительский HTTP шлюз (раздел 6.6.5 на стр. 84);
  - `Порт родительского прикладного шлюза` указывает порт, на котором слушает «родительский» HTTP-шлюз (раздел 6.6.5 на стр. 84);
  - `Учетное имя анонимного пользователя` задает логин анонимного пользователя
  - `Пароль анонимного пользователя` задает пароль для ftp-логина (раздел 6.6.7 на стр. 85);
  - `Режим доверяемого назначения (в прозрачном режиме)` включает режим доверяемого назначения для прозрачного режима (раздел 6.6.4 на стр. 84);
  - `Пропускать некорректные ответы` включает режим пропускания некорректных ответов (раздел 6.6.1.3 на стр. 82);
  - `Допускать многострочные запросы` включает режим исправления многострочных запросов (раздел 6.6.1.3 на стр. 82);
  - `Список блокируемых URI` задает список блокируемых URI;
  - `URL сервера адаптации контента (ICAP)` задает хост антивирусного сервера. Если хост указывается, то включается режим антивирусного сканирования (раздел 13 на стр. 165);
  - `Включить NTLM-аутентификацию` включает поддержку NTLM
  - `Контролер NT-домена` определяет имя контролера NT домена для NTLM;
  - `Резервный контролер NT-домена` определяет имя резервного контролера NT домена для NTLM;
  - `Имя NT-домена` определяет имя NT-домена для NTLM;
  - `Исходящий адрес` — указывает адрес, с которого будет осуществляться доступ с прикладного шлюза для данной группы.
    Т.е. можно определить, с какого именно IP-адреса МЭ (если их несколько) шлюз будет устанавливать соединения с внешними ресурсами для определенного списка доступа.
  - `Использовать авто-определение mime на HTTP` включает автоматическое определение mime-типа файла.
- `block-type mime` содержит имена блокируемых MIME-типов;
- `trans-mime` содержит соответствие MIME типов заглушкам (раздел 6.6.2 на стр. 83);
- `Блокируемые команды` содержит настройки блокировки HTTP-команд.

Фильтрация по IP-адресу/порту назначения указывается, как и у остальных прикладных шлюзов, в разделе `Исходящие правила` списка доступа. В поле `протокол` для HTTP-запросов устанавливается значение «http», для ftp-запросов — «ftp», «ssl» — для запросов установления SSL туннеля.

Фильтрация по URL позволяет блокировать сайты по именам и конкретные подразделы сайтов, а также позволяет делать фильтрацию с подменой контента

Фильтрация по URL задается в виде алиаса в опциях списка доступа. `Список блокируемых URI` представляет собой набор фильтров, где ключом является регулярное выражение, определяющее

блокируемый URL, а значением — режим блокировки. Режим «BLOCK» включает блокировку, режим «WARNING» — предупреждение, остальные режимы предназначены для подмены контента и описаны в разделе 6.6.2 на стр. 83. Запрос на заблокированный в режиме «BLOCK» URI считается нарушением политики безопасности. В режиме «WARNING» пользователю разрешается доступ к запрошенному ресурсу, но в журнал пишется предупреждение.

Фильтрация по mime-type позволяет заблокировать определенные mime-типы ответов.

Примечание Указанный сервером mime-тип является носит рекомендательный характер для броузера. Многие броузеры игнорируют указанный сервером MIME-тип или пытаются подкорректировать его в соответствии ожидаемым типом ответа. В следствие этого, многие серверы неверно сообщают MIME-тип ответа. Например, для подключаемых файлов javascript вместо ожидаемого MIME-типа «application/javascript» зачастую приходит ответ типа «text/plain» или «text/html».

Фильтрация по MIME-типу задается в разделе `block-type` списка доступа.

6.6.1.2.1 Блокирование команд HTTP–протокола

Прикладной шлюз HTTP позволяет блокировать отдельные команды HTTP-протокола. Набор заблокированных команд задается во вкладке `Блокируемые команды` списка доступа.

Реально на практике используются только пять основных команд HTTP-протокола:

- `GET` для получения страниц;
- `POST` для посылки форм;
- `HEAD` для получения заголовков страницы (используется некоторыми старыми http-клиентами);
- `CONNECT` для туннелирования протокола SSL.

Остальные команды применяются в только в специальных случаях.

6.6.1.3 Дополнительные опции прикладного шлюза HTTP

6.6.1.3.1 Режим пропускания некорректных ответов

Некоторые HTTP-серверы отсылают в Интернет ответы без строки состояния и HTTP-заголовков. Как правило, это происходит при некорректно написанных CGI-скриптах или на плохо разработанных HTTP-серверах. К сожалению, броузеры и некоторые популярные HTTP-шлюзы (такие как `Squid`) не считают такие ответы ошибочными и обрабатывают их. По умолчанию прикладной шлюз HTTP не пропускает такие ответы, считая их некорректными. Режим пропускания некорректных ответов включается в опции `Пропускать некорректные ответы`. Мы рекомендуем не включать этот режим без реальной необходимости.

Включая режим пропускания некорректных ответов, вы лишитесь:

- проверок HTTP-заголовков для ответов с некорректной строкой статуса;
- автоматического определения MIME-типа для некорректных ответов;
- блокировки по MIME-типу для некорректных ответов.

6.6.1.3.2 Режим выправления многострочных запросов

Как было обнаружено, в некоторых случаях браузер Netscape Communicator 4 посылает запросы с переносом строки в URL. Это происходит в случае, если URL был сформирован таковым из JavaScript. Такое поведение является грубым нарушением HTTP-протокола и потому не воспринимается некоторыми веб серверами и нашим прикладным шлюзом по-умолчанию. Включая опцию списка доступа `Допускать многострочные запросы`, вы разрешаете шлюзу определять и исправлять запросы,

содержащие один лишний перенос строки в запросе. Мы рекомендуем включать эту опцию только для определенных клиентов и при наличии реальной необходимости.

6.6.1.3.3 Настройка транспортных сессий

Настройка транспортных сессий протокола HTTP 1.1 задается в разделе `Транспортные сессии`. При этом задаются следующие параметры:

Максимально допустимое время между запросами (в секундах) — по истечении этого времени сессия разрывается.

Разрешить работу с транспортными сессиями HTTP — разрешение или запрещение на работу с постоянными соединениями, т.е. транспортными сессиями протокола HTTP версии 1.1.

Максимальное количество запросов за транспортную сессию — при превышении указанного количества сессия разрывается.

6.6.2 Подмена контента и блокировка баннеров

Прикладной шлюз HTTP содержит механизм блокировки запросов с подменой контента, используемый, как правило, для блокировки баннеров.

В отличие от простой блокировки, при блокировке баннеров возникает необходимость в подмене контента, поскольку броузеры и javascript-программы некорректно работают в случае получения странички об ошибке доступа вместо ожидаемых изображений и текстов.

Мы рекомендуем подменять контент для картинок на предложенные нами изображения и на пустой текстовый файл для javascript, текста и html.

Прикладной шлюз HTTP не может самостоятельно определить, ответ какого типа нужно отдать пользователю в ответ на заблокированный запрос. В `Списке блокируемых URI` можно указать MIME-тип ответа для отдельных блокируемых URL, либо выбрать режим «AUTODETECT» — автоматического определения.

В режиме «AUTODETECT» прикладной шлюз делает запрос на сервер и ожидает ответа. После получения заголовков прикладной шлюз обрывает связь с сервером (не получая при этом тела ответа) и возвращает пользователю заглушку. В режиме «AUTODETECT» прикладной шлюз пропускает ответы с HTTP-Redirect.

Возвращаемые файлы-заглушки настраиваются закладке «trans-mime»

6.6.3 Шаблон страницы об ошибке, генерируемой прикладным шлюзом HTTP

Шаблон страницы, генерируемой прикладным шлюзом в случае возникновения какой-либо ошибки, находится в каталоге `/opt/fw/share` в файле `error_text`.

Шаблон представляет собой HTML-код страницы с внедренным в него набором ключевых слов. При генерации страницы ключевые слова заменяются на сообщения, описывающие причину произошедшего сбоя.

В шаблоне допустимо применять следующие ключевые слова:

%message% — причина ошибки
%code% — номер ошибки
%error% — название ошибки
%proxy% — текущая версия прикладного шлюза

Внешний вид страницы об ошибке можно изменить. По умолчанию файл `error_text` содержит следующий код:

```
<body bgcolor=white>
<center><table width="80%">
<tr bgcolor=black><td>
<table width="100%" cellspacing=0 cellpadding=5><tr bgcolor="#8080A0"><td>
<h1>%code% %message%</h1>
</td></tr><tr bgcolor=white><td>
<p>%error%</p>
<em>Generated by Z-2 Firewall version %proxy%></em>
</td></tr></table></table>
</center>
</body>
```

6.6.4 Прозрачный режим

Прикладной шлюз HTTP может быть использован в прозрачном режиме. При этом есть две различных схемы функционирования прикладного шлюза — обычный режим и режим *доверяемого назначения*. Режим выбирается в опции доступа `Режим доверяемого назначения (в прозрачном режиме)`.

Протокол HTTP выгодно отличается от других протоколов тем, что позволяет определить сервер назначения по URI, полностью содержащемся в прикладном запросе.

В обычном режиме прикладной шлюз HTTP игнорирует оригинальный IP-адрес назначения и определяет адрес назначения по прикладному запросу. Именно так функционирует прикладной шлюз HTTP в непрозрачном режиме.

В режиме доверяемого назначения прикладной шлюз HTTP использует оригинальный адрес назначения и передает по нему http-запрос. Таким образом сокращается время отклика за счет того, что прикладному шлюзу более не требуется делать DNS-запросы для определения адреса сервера. В этом режиме пользователи могут посылать http-запросы на определенный URI к серверам, не обслуживающим данный URI, а так же использовать внешние прикладные шлюзы протокола HTTP.

Примечание В «непрозрачном» режиме не следует выставлять режим доверяемого назначения, поскольку это приводит только к понижению производительности за счет ненужных запросов к пакетному фильтру.

«Прозрачный» режим работы прикладного шлюза HTTP реализуется только совместно с пакетным фильтром. Для его реализации пакетный фильтр должен быть настроен так, чтобы выделять пакеты HTTP-протокола и переадресовывать их для обработки в прикладной шлюз.

6.6.5 Цепочки http–шлюзов

Http-протокол допускает создание цепочки http-шлюзов. Прикладной шлюз HTTP умеет функционировать в таких цепочках с любыми поддерживающими HTTP прикладными шлюзами.

Родительский HTTP шлюз указывается в опциях `Хост родительского прикладного шлюза` и `Порт родительского прикладного шлюза` списка доступа. В случае, когда переназначение не требуется, значение `Хост родительского прикладного шлюза` и `Порт родительского прикладного шлюза` выставляются в «none».

В разделе `noproxy` указываются имена хостов, запросы на которые передаются непосредственно, игнорируя «родительский» шлюз. Проверка имен происходит с именем хоста в URI.

6.6.6 Аутентификация пользователей

Прикладной шлюз HTTP поддерживает аутентификацию пользователей через сервер аутентификации. Из-за ограничений http-протокола поддерживаются не все возможные PAM-модули, а только обеспечивающие обмен паролями по схеме логин/пароль (например, `pam_db`). Более сложные схемы не поддерживаются (например, `pam_skey`). Аутентификация пользователей возможна только в «непрозрачном» режиме работы прикладного шлюза.

Http — протокол без поддержки сессий, таким образом, авторизация и проверка пароля происходят на каждый запрос. Для уменьшения нагрузки на сервер аутентификации прикладной шлюз HTTP кеширует auth-запросы в течении минуты.

Необходимость аутентификации пользователей задается в опциях списка доступа (раздел 6.6.1.2 на стр. 81). Настройка IP-адреса и порта сервера аутентификации осуществляется в разделе `Сервер аутентификации`.

6.6.7 FTP–модуль

Прикладной шлюз HTTP поддерживает запросы к FTP URI. В списке доступа необходимо указать адреса и порты, на которые возможно открытие FTP-соединения. Порт задается только для канала команд FTP, соединения для канала данных открываются в обход правил доступа, но только на сервер, с которым открыто соединение команд. Выходящие соединения проверяются по записям с установленным значением «ftp» поля «протокол» раздела `Исходящие правила` списка доступа.

По умолчанию FTP-модуль аутентифицируется на сервере с анонимным логином и паролем, который задается в списке доступа в опциях «Учетное имя анонимного пользователя» и «Пароль анонимного пользователя».

В случае использования «родительского» шлюза FTP-запросы передаются к нему по протоколу HTTP.

FTP-серверы не сообщают пользователям MIME-типы передаваемых файлов. В разделе `trans-mime` задается имя файла, содержащего правила определения MIME-типа файла по его имени. Файл правил трансляции создается с помощью любого текстового редактора. Базовый файл трансляции имен входит в состав МЭ «Z-2»

6.6.7.1 Шаблон ftp–страницы

Шаблон ftp-страницы находится в каталоге `/opt/fw/share в файле ftp_list`. Шаблон представляет собой HTML-код с внедренным в него набором ключевых слов. При генерации страницы ключевые слова заменяются на следующие данные:

%message% — описание текущего каталога, полученное от ftp-сервера
%host% — имя хоста, к которому производится соединение
%dir% — текущий каталог
%proxy% — текущая версия прикладного шлюза

Структурно шаблон состоит из трех частей:

1. заголовок страницы — HTML-код, который вставляется до списка файлов и каталогов;
2. строка «%%%» — она заменяется на список файлов и каталогов;
3. окончание («подвал») страницы — HTML-код, который вставляется после списка файлов и каталогов.

Внешний вид ftp-страницы можно изменить. По умолчанию файл `ftp_list` содержит следующий код:

```
<!-- HTML listing generated by http-gw version %proxy% -->
<html><head>
<title>FTP directory listing of ftp://%host%</title>
</head>
<body bgcolor=white>
<center>
<table width="80%">
<tr bgcolor=black><td>
<table width="100%" cellspacing=0 cellpadding=5><tr bgcolor="8080A0"><td>
<h1>FTP directory <a href="ftp://%host%/%curdir%">%host%/%curdir%</a>
<pre>%message%</pre>
</h1></td></tr><tr bgcolor=white><td><pre>
%%%
</pre><tr bgcolor="#8080A0"><td>
<em>Generated by http-gw version %proxy%
</em></td></tr></table>
</center>
</center>
</body></html>
```

6.6.8 Туннелирование протокола SSL

Прикладной шлюз HTTP поддерживает туннелирование обмена по протоколу SSL (https, CONNECT-метод протокола HTTP).

Право на доступ по протоколу SSL проверяется по записям с установленным значением «ssl» поля «протокол» раздела `Исходящие правила` списка доступа. Обычно разрешают доступ по протоколу SSL только на 443 порт.

Запросы, передаваемые по SSL туннелю, зашифрованы и, таким образом, их контроль шлюзом не осуществляется.

Для работы SSL-туннеля вы должны также разрешить пользователю использование команды CONNECT (раздел 6.6.1.2.1 на стр. 82).

В случае использования родительского прикладного шлюза все CONNECT-запросы передаются ему по HTTP.

6.7 Прикладной шлюз протокола SMTP

Прикладной шлюз SMTP предназначен для осуществления, контроля и разграничения обмена электронной почтой по протоколу SMTP (Simple Mail Transfer Protocol).

Прикладной шлюз SMTP представляет собой комплекс из двух программ — `smtp-gw`, резидентного процесса, принимающего сообщения по SMTP-протоколу и сохраняющего их на диск, `smtp-fwdr` — почтового ретранслятора, передающего сохраненную почту программе sendmail для дальнейшей обработки.

`Smtp-gw` осуществляет:

- прием сообщений по протоколу SMTP;
- проверку допустимости передачи письма по набору правил, учитывающих почтовые адреса отправителя и получателя и IP-адрес отправителя;
- проверку IP-адреса отправителя через доступные в интернет (по протоколу ORBS) базы данных почтовых маршрутизаторов — отправителей спама;
- проверку допустимости почтового обмена в соответствии с SMTP-протоколом;
- аутентификацию пользователей на сервере аутентификации.

`Smtp-fwdr` осуществляет:

- проверку почтовых сообщений на корректность;
- ретрансляцию почтовых сообщений программе *sendmail*;
- (опционально) передачу сообщения на антивирусную проверку (раздел 13 на стр. 165).

`Smtp-fwdr` запускается с определенной периодичностью системным планировщиком задач `CRON`. Все необходимые для этого настройки выполняются администратором межсетевого экрана через графический интерфейс (смотри книгу *«Универсальный графический интерфейс 2.4 Руководство администратора»*).

Программы `smtp-gw` и `smtp-fwdr` используют общий каталог для хранения писем и поддерживают параллельный доступ к нему, таким образом, возможна одновременная работа `smtp-gw` и `smtp-fwdr`, а также одновременная работа нескольких smtp-gw/smtp-fwdr над одним каталогом. Каталог не следует размещать на NFS-разделах.

Прикладной шлюз осуществляет проверку используемых почтовых адресов на допустимость. Допустимым считается почтовый адрес, соответствующий формату, определенному в RFC 822 без роутинговой информации, UUCP-адресации и других редко используемых в сети Интернет возможностей RFC 822.

6.7.1 Настройки прикладного шлюза SMTP

Управление функционированием прикладным шлюзом SMTP осуществляется с помощью:

1. настроек, управляющих работой резидентного процесса (раздел 6.7.1.1 на стр. 87);
2. установки пути к каталогу для сохранения почтовых сообщений (раздел 6.7.1.3.1 на стр. 89);
3. установки максимально допустимого размера почтового сообщения (раздел 6.7.1.3.2 на стр. 89);
4. настроек сервера аутентификации (`Сервер аутентификации`) (раздел 6.7.1.1 на стр. 87);
5. правил фильтрации (раздел 6.7.1.2 на стр. 88).

6.7.1.1 Настройки резидентного процесса

Прикладной шлюз SMTP работает по «forked» — схеме, т.е. изначально в системе запускается один процесс smtp-gw, ожидающий соединения по smtp-протоколу. При приходе запроса создается новый процесс smtp-gw, которому передается запрос. Одновременно один процесс обслуживает одного

клиента, при окончании smtp-диалога процесс завершается. Подробно настройки описаны в главе 6.1.1 на стр. 70.

Лимит на число процессов следует подбирать исходя из мощности машины и интенсивности почтового потока. По SMTP-протоколу сбой при установлении соединения не считается фатальной ошибкой. Таким образом, задание небольшого лимита позволяет распределить нагрузку на сервер.

Рекомендуемая величина максимального числа процессов — от 10 до 512.

6.7.1.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к прикладному шлюзу протокола `smtp` производится с помощью раздела `Политика`. Список доступа привязан к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Список доступа прикладного шлюза SMTP состоит из:

- `Входящие правила` — правила контроля входящих соединений (раздел 6.2 на стр. 73);
- `Опции шлюза` — опции правила:
  - `Политика протокола SMTP` определяет политику фильтрации писем на основе почтовых адресов отправителя/получателя (раздел 6.7.4 на стр. 91);
  - `Политика для аутентифицированных пользователей` определяет политику фильтрации писем на основе почтовых адресов отправителя/получателя для аутентифицированных пользователей (раздел 6.7.5 на стр. 94);
  - `Включить SMTP-аутентификацию` разрешает SMTP-аутентификацию;
  - `Зона ORBS` определяет используемый ORBS-сервер (раздел 6.7.6 на стр. 95);
  - `Ответ ORBS` определяет настройку ORBS сервера;
  - `Действие, применяемое к спаму` определяет настройку ORBS сервера;
  - `Включить проверку DNS` — если данный пункт выбран, прикладной шлюз SMTP производит проверку на наличие доменной части почтового адреса отправителя в службе DNS как адреса или почтового сервера для этого домена. Если указанное отправителем доменное имя в DNS не найдено, почтовое сообщение с такого адреса отвергается.
  - `Не ограничивать число RCPT TO` — Выключить проверку ограничения на число получателей письма. Это иногда бывает необходимо для работы с некорректными программами груповой рассылки
  - Включить Greylisting — включить опцию блокировки спама методом Greylisting
  - Ограничение на количество соединений с одного IP — Установить число максимального количества соединений с одного IP-адреса
  - Отключить запись Received заголовка в письмо — По умолчанию шлюз smtp добавляет в заголовки принятых писем строку «Received: from ...». Если включить эту опцию, то строка добавляться не будет.
  - `Строка приветствия шлюза` — установить «собственную» строку приветствия шлюза. Строка приветствия выдается клиенту, отправляющему почту, после установления соединения. (По умолчанию шлюзом выдается строка «Z-2 Firewall ВЕРСИЯ SMTP proxy ready to service»).
- `Журналируемые команды` — список команд протокола SMTP, которые будут регистрироваться в протоколах работы МЭ «Z-2»;
- `smtp-to` — данные, необходимые для фильтрации на основе адресов электронной почты получателя (раздел 6.7.4 на стр. 91);
- `smtp-from` — данные, необходимые для фильтрации на основе адресов электронной почты отправителя.

6.7.1.3 Дополнительные настройки прикладного шлюза SMTP

6.7.1.3.1 Настройки каталога передачи почты

Каталог передачи почты задается в разделе `Spool-каталог` конфигурации шлюза. В этом разделе определяется единственный параметр: `Имя spool-каталога` — путь к каталогу, в который будут записываться принятые почтовые сообщения.

6.7.1.3.2 Настройки обработки почты

Настройки обработки почты задаются в разделе `Обработка почты`. С помощью данного раздела определяются следующие параметры:

- `Максимальная длина сообщения (в байтах)` — максимально допустимый размер почтового сообщения.
  Если почтовое сообщение превышает этот лимит, то такое письмо не переправляется адресату, а уничтожается. Серверу-отправителю подобного почтового сообщения формируется код с ошибкой приема письма `550 Size exeeded`.
  Если значение параметра `Максимальная длина сообщения (в байтах)` установлено в ноль, то ограничение на размер передаваемых писем не вводится.
- `Файл базы данных Greylisting` — файл, в котором хранится база данных триплетов. Файл должен быть различным для различных экземпляров smtp-шлюзов —
- `Начальная задержка Greylisting` — задержка, в течении которой письмо с неизвестным триплетом отвергается сервером. Величина задается в секундах. Рекомендуемое значение — 1 час. Меньшие значения способны сократить задержки при доставке почты, но снижают эффективность метода.
- `Время жизни новых записей Greylisting` — время, в течении которого ожидается повторная попытка отправки письма. Задается в секундах. Рекомендуемое значение — 4 часа. Меньшие значения способны сократить объем базы данных сервера, но могут привести к увеличению задержек в доставке почты.
- `Время жизни активных записей Greylisting` — время, в течении которого сохраняются триплеты доставленных писем. Триплет удаляется если в течении указанного времени по нему не проходило ни одного письма. Задается в секундах. Рекомендуемое значение — 1 месяц. Меньшие значения способны сократить объем базы данных сервера, но могут привести к увеличению задержек в доставке почты.

6.7.1.3.3 Настройки сервера аутентификации

Настройки сервера аутентификации задаются в разделе `Сервер аутентификации`. В этом разделе определяется единственный параметр: `Порт сервера аутентификации` — имя порта, на котором сервер аутентификации принимает соединения.

По умолчанию в качестве сервера аутентификации используется локальный хост (127.0.0.1), а в качестве порта - auth-server (999).

6.7.2 Настройки почтового ретранслятора `smtp-fwdr`

Настраиваемыми являются следующие параметры:

1. путь к каталогу, где хранятся почтовые сообщения, успешно принятые прикладным шлюзом SMTP;
2. путь к каталогу для сообщений с некорректной структурой;
3. путь к программе *sendmail*, используемой для доставки почты;
4. указания адреса антивирусного сервера (раздел 13 на стр. 165).

Настройка spool-каталога Настройка путей к каталогам и относящимся к ним опций. С его помощью настраиваются следующие параметры:

Имя spool-каталога — полный путь (без символа «/» в конце) к каталогу, в котором хранятся почтовые сообщения, принятые smtp-gw.

Каталог для некорректных писем — полный путь (без символа «/» в конце) к каталогу, куда smtp-fwdr будет записывать сообщения, не отвечающие критериям фильтрации или отвергнутые почтовым сервером.

Максимальное время нахождения сообщения в очереди (в секундах) — максимально допустимое время (в секундах) нахождения почтового сообщения в очереди на доставку. По истечении этого времени почтовое сообщение перемещается в каталог с «некорректной» информацией.

Конфигурация ретрансляции почты — установки проверки сообщений.

URL сервера адаптации контента (ICAP) — задает имя хоста антивирусного сервера. Значение none выключает проверку сообщений на вирусы.

Конфигурация программы доставки почты — указание на используемый почтовый сервер. Содержит единственный параметр — Исполняемый файл sendmail. Его значением является полный путь к исполняемому файлу программы sendmail, которая используется для доставки адресату прошедших все проверки почтовых сообщений.

Программа sendmail должна быть соответствующим образом настроена. Рекомендации по настройке sendmail приведены в разделе 6.7.3 на стр. 90.

Для периодического запуска Почтового ретранслятора необходимо выполнить настройку планировщика задач CRON. Это может быть выполнено с помощью раздела Планировщик задач графического интерфейса (более подробно работа с планировщиком задач описана в книге *«Универсальный графический интерфейс 2.4 Руководство администратора»*).

6.7.3 Настройка почтовой системы для работы с прикладным шлюзом протокола SMTP

Для совместной работы с прикладным шлюзом протокола SMTP программа, обрабатывающая почту, должна быть соответствующим образом сконфигурирована. В данном разделе описывается процедура настройки почтовой службы на основе программы sendmail.

Для корректной работы связки smtp-qw — sendmail необходимо, чтобы программа sendmail была запущена в режиме демона, но при этом не «слушала» на порту. Чтобы добиться этого, необходимо в файл /etc/default/sendmail поместить строку «MODE=»:

```
echo "MODE=" >> /etc/default/sendmail
```

Далее необходимо включить в sendmail поддержку mailertable и создать таблицу маршрутов доставки почты. Для этого необходимо выполнить следующие операции.

1. Перейдите в каталог с конфигурационным файлом программы sendmail:

   ```
   cd /usr/lib/mail/cf
   ```

2. Создайте копию .mc-файла, на основе которого был создан конфигурационный .cf файл, используемый sendmail данный момент.

   ```
   cp main-v7sun.mc main-v7sun-smtp_gw.mc
   ```

   Примечание Все изменения конфигурации далее будут производиться в новом .mc файле, а старый должен остаться без изменения, чтобы в случае каких-либо проблем можно было вернуться к предыдущим настройкам.

3. Добавьте в новый .mc поддержку mailertable. Это можно сделать следующими командами:

   ```
   echo "FEATURE(\'mailertable', \'dbm /etc/mail/mailertable')dnl" >> main-v7sun-smtp_gw
   echo "define(\'DATABASE_MAP_TYPE', \'dbm')dnl" >> main-v7sun-smtp_gw.mc
   ```

4. Создайте новый файл конфигурации sendmail.cf на основе исправленного .mc файла:
   ```
   /usr/ccs/bin/m4 ../m4/cf.m4 main-v7sun-smtp_gw.mc > sendmail.cf
   ```
5. Сохраните текущий файл sendmail.cf и установите новый, с поддержкой mailertable:
   ```
   cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.before.z2
   cp sendmail.cf /etc/mail/sendmail.cf
   ```
6. Создайте файл /etc/mail/mailertable следующего содержания:
   ```
   my.domain.ru              esmtp:[host.my.domain.ru]
   some.domain.ru   esmtp:[192.168.111.2]
   ```
   Где my.domain.ru и some.domain.ru — это домены, на которые пересылается почта после обработки прикладным шлюзом SMTP. Например, запись my.domain.ru esmtp:[smtp.intra.net] означает, что для домена my.domain.ru необходимо пересылать почту на сервер smtp.intra.net посредством протокола SMTP; some.domain.ru esmtp:[192.168.111.2] — для домена some domain.ru пересылать почту на сервер с адресом 192.168.111.2 посредством протокола SMTP.
   Примечание Квадратные скобки означают работу по A-записям в DNS, а не по MX-записям. Таким образом, почта перенаправляется на конкретный сервер.
7. Сгенерируйте на основе /etc/mail/mailertable базу данных маршрутов, доступную программе sendmail:
   ```
   /usr/sbin/makemap dbm  /etc/mail/mailertable < /etc/mail/mailertable
   ```
8. Перезапустите sendmail для того, чтобы новые настройки вступили в силу:
   ```
   /etc/init.d/sendmail stop
   /etc/init.d/sendmail start
   ```

В случае необходимости можно отменить данные настройки, вернув на место сохраненный файл конфигурации sendmail.cf и снова перезапустив почтовую службу.

Более подробную информацию по настройке программы sendmail можно найти в /usr/lib/mail/README.

6.7.4 Фильтрация писем на основе почтовых адресов

В ходе SMTP-диалога прикладной шлюз SMTP осуществляет проверку письма по правилам приема сообщений, задаваемым в текущем списке доступа, привязанном к данному клиенту по комбинации исходящий IP-адрес/порт. Списки состоят из набора правил проверки исходящего почтового адреса, почтового адреса назначения и политики безопасности «по умолчанию».

Политика «по умолчанию» задается во вкладках `smtp-from` и `smtp-to` списка доступа `Политика`:

- `RELAY` и `HPRELAY` задают политику *черного списка*, письма принимаются в случае если они явно не заблокированы в правилах проверки адресов. Обычно данную политику выбирают для списков, соответствующих локальным сетям, для разрешения посылки сообщений во внешний мир. Письма, проходящие по политике `HPRELAY`, являются высокоприоритетными и обрабатываются вне очереди.
- `OK` задает политику *белого списка*, письма принимаются только если они явно разрешены в правилах проверки адресов. Обычно данную политику выбирают для списков, соответствующих внешним сетям, которым разрешено посылать почту только на локальные домены.
- `REJECT` при этой политике любые письма отвергаются с кодом ошибки 550, списки проверки по адресам не используются. Эта политика используется для блокировки приема писем с конкретных почтовых серверов.

По ходу получения SMTP-команд `MAIL FROM` и `RCPT TO` производятся соответствующие проверки адресов. После проверки выносится решение о приеме письма. В случае отрицательного решения проверки smtp-gw отказывается принимать письмо, в результате чего посылающий его SMTP-сервер

формирует информационное сообщение об этом пользователю. В случае, если решение становится известно на момент получения MAIL FROM, об этом сообщается посылающему серверу. В остальных случаях решения сообщаются в ответ на команду RCPT TO.

Правила приема сообщений задаются в двух разделах списка доступа — в разделе smtp-from задаются правила проверяющие почтовый адрес отправителя SMTP конверта, smtp-to — проверяющие почтовый адрес получателя.

Ключом поиска правила является шаблон почтового адреса, значением — решение проверки.

Почтовые адреса smtp-конверта должны быть *допустимыми*, перед проверками происходит процедура их канонизации.

На каждой проверке ищется наиболее подходящее для данного письма правило, и оно исполняется. Другие правила не рассматриваются. Порядок следования правил таким образом не важен.

Шаблон задания почтового адреса:

username@domainname индивидуальный почтовый адрес, например:

dolg@mail.ru

@domainname любой пользователь из указанного домена, например:

@mail.ru

domainname любой пользователь из указанного домена и всех его поддоменов, например:

ru

Действия по доставке почтового сообщения:

- REJECT отказать в приеме письма. При получении REJECT по любой из проверок письмо отвергается независимо от другой проверки и политики по-умолчанию.
- RELAY принять письмо.
- HPRELAY принять письмо вне очереди.
- OK прекратить данную проверку и перейти к следующей.

Для пропускания письма при политике *черного списка* (RELAY) достаточно, чтобы письмо не получило REJECT по проверке адресов. Действия RELAY и OK при такой политике не различаются.

При политике *белого списка* (OK) письмо должно получить RELAY по хотя бы одной из проверок адресов и не получить REJECT ни по какой другой.

В общем случае логика прикладного шлюза SMTP построена так: положительное решение о приеме письма принимается в случае, если нет ни одного REJECT-решения и есть хотя бы одно RELAY-решение среди трех значений: значения по умолчанию и результатов проверок по почтовому адресу отправителя и адресу получателя сообщения.

6.7.4.1 Примеры применения правил фильтрации

6.7.4.1.1 Разрешение ретрансляции писем для почтового сервера локальной сети

Для разрешения серверу отправлять почту через прикладной шлюз SMTP следует добавить список доступа для этого сервера с политикой *RELAY*. Вторым вариантом решения этой задачи является добавление правила во вкладке smtp-from в любом списке доступа, куда попадает этот сервер (в нашем случае SMTP-сервер с адресом 172.16.10.2), см. рисунок 6.5.

6.7.4.1.2 Разрешение ретрансляции писем на локальные домены

Для ретрансляции писем на локальные домены следует задать список, разрешающий соединение из внешних сетей с политикой доступа (раздел Политика) *OK*. Во вкладке smtp-to списка за-

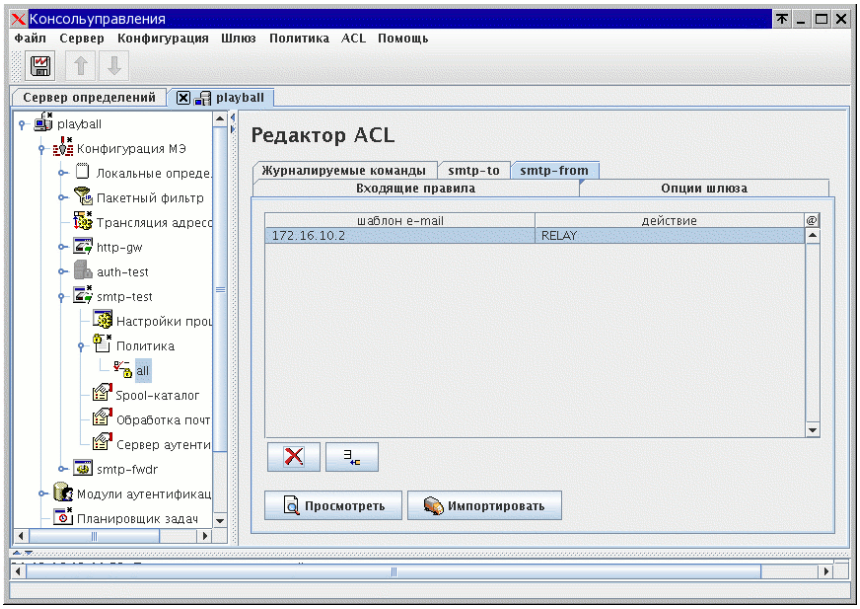

Рис. 6.5: Разрешение ретрансляции писем для почтового сервера

дать локальный домен с действием RELAY. В этом примере позволено принимать любые письма для почтового домена my-domain.ru и любых его поддоменов (рисунок 6.6).

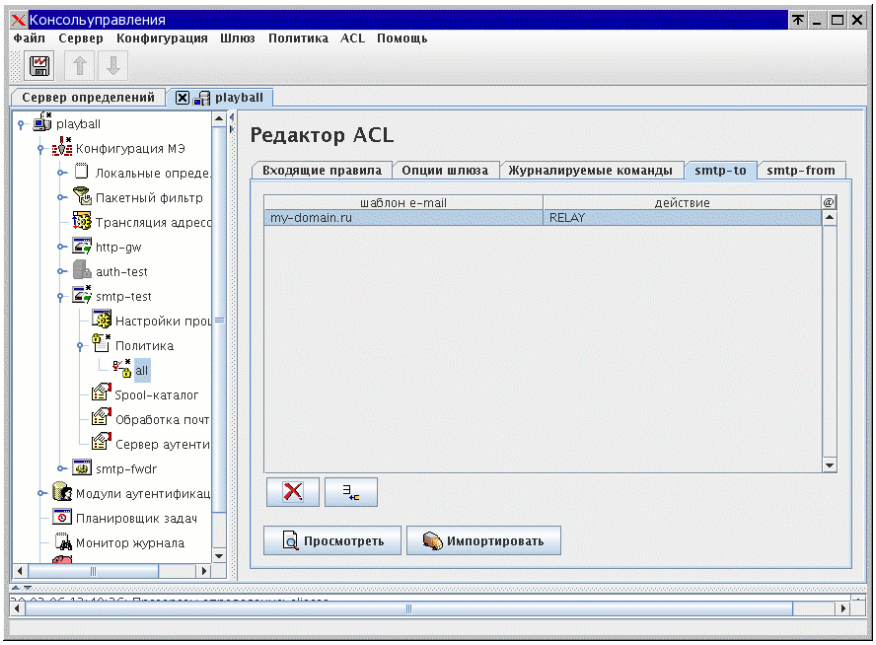

Рис. 6.6: Разрешение ретрансляции писем на домен my-domain.ru

6.7.4.1.3 Блокировка нежелательных отправителей по IP–адресу

Необходимо создать для них список доступа для нежелательных отправителей и назначить политику по умолчанию в REJECT или, например, в 550 We do not accept email from spammers.

6.7.4.1.4 Блокировка нежелательных отправителей по почтовому адресу

При этой блокировке необходимо учитывать, что адрес отправителя в электронной почте — значение информативное и указывается отправителем. Адреса задаются во вкладке smtp-from списка доступа.

6.7.4.1.5 Разрешение ретрансляции писем для локальной сети с проверкой from

Вместо полного разрешения ретрансляции почты из локальной сети можно разрешить эту ретрансляцию только исходящими локальными почтовыми адресами (в дополнение к проверке исходящего IP-адреса).

Для этого нужно сделать список доступа для локальных пользователей с политикой *белого списка* и добавить RELAY правило во вкладке smtp-from для локального почтового домена

6.7.5 Аутентификация пользователей

Прикладной шлюз SMTP поддерживает аутентификацию пользователей через сервер аутентификации. Из-за ограничений SMTP-протокола поддерживаются не все возможные PAM-модули, а только обеспечивающие обмен паролями по схеме логин/пароль (например pam_db).

SMTP-аутентификация происходит опционально в случае, если ее поддерживает почтовый клиент/MTA. Smtp-gw позволяет использовать разные политики приема почты для обычных и аутентифицированных пользователей, а также позволяет запретить прием почты от неаутентифицированных пользователей.

Необходимость аутентификации пользователей задается в опциях списка доступа (раздел 6.7.1.2 на стр. 88). Настройка IP-адреса и порта сервера аутентификации осуществляется в разделе `Сервер аутентификации`.

6.7.5.1 Примеры применения правил фильтрации с аутентификацией

6.7.5.1.1 Разрешение ретрансляции писем на локальные домены и разрешение ретрансляции на любые после аутентификации

Для ретрансляции писем на локальные домены следует задать список, разрешающий соединение с любых адресов с политикой *белого списка*. В разделе `smtp-to` списка задать локальный домен с действием `RELAY`. Далее следует включить SMTP-аутентификацию в опциях правила и выбрать политику *черного списка* для аутентифицированных пользователей.

6.7.5.1.2 Разрешение ретрансляции писем для локальной сети после аутентификации

Для разрешения локальным пользователям отправлять почту через SMTP-GW следует добавить список для локальных пользователей с политикой *черного списка* для аутентифицированных пользователей и политику `REJECT` для остальных.

6.7.6 Проверка письма через ORBS

Прикладной шлюз SMTP умеет взаимодействовать с базами данных блокировки IP-адресов отправителей, доступными по протоколу ORBS.

В опциях шлюза списка доступа указывается имя хоста ORBS сервера, ожидаемый от него ответ (его необходимо узнать у администратора ORBS сервера, для публичных ORBS серверов в интернет эта информация обычна указана на веб-сайте), действие в случае положительного ответа ORBS-сервера. Действие `отвергать` означает отказ в принятии письма и равносильно REJECT-политике по-умолчанию. Действие `отмечать в заголовке` означает что письмо будет передано (если пройдет проверки по базе данных релеинга), но в его заголовки будет добавлено предупреждение о сбое ORBS проверки `X-ORBS-Warning: Possible SPAM`. В этом случае возможна последующая фильтрация писем прикладными средствами (например, можно откладывать такие письма в отдельную папку почтового клиента).

Дополнительная информация о ORBS может быть получена по адресу http://www.mail-abuse.org.

6.7.7 Проблема открытых почтовых ретрансляторов и рассылки спама

Почтовый сервер, позволяющий прием и отправку любых почтовых сообщений из внешних сетей, называется *открытым почтовыми ретрансляторам* (*open relay*). Подобная конфигурации сервера наносит вред пользователям сети Интернет, а также непосредственно сети пользователя.

Рано или поздно, адрес подобного сервера становится известен организациям, занимающимся рассылкой «спама» — непрошенной корреспонденции рекламного характера. Как правило, это происходит самое позднее через месяц жизни открытого ретранслятора — для поиска подобных серверов используются специализированные программы. Далее через открытый ретранслятор начинается рассылка спама. Подобная рассылка осуществляет серьезную нагрузку на канал передачи данных и на почтовый сервер. В дальнейшем открытый ретранслятор считается источником спама и может быть

заблокирован администраторами других почтовых серверов и службами противодействия распространения спама.

В связи с этим необходимо ввести ограничения на доставку почтовых сообщений из внешних сетей, сделав возможным передачу почты только на Ваш почтовый домен, как это описано в разделе 6.7.4 на стр. 91.

При наличии удаленных пользователей мы рекомендуем следующие схемы работы:

- Использование удаленными пользователями SMTP-сервера их Интернет-провайдера. Подобная практика является наилучшим решением проблемы внешних пользователей.
- Использование SMTP-аутентификации для внешних пользователей.
- Использование средств кодирования потоков (VPN) для доступа внешних пользователей к корпоративному серверу.

6.7.8 Блокировка спама методом Greylisting

Greylisting — метод борьбы с спамом и почтовыми вирусами, работающий на уровне протокола SMTP. Он основан на особенностях передачи писем по протоколу SMTP. Метод работает без анализа содержимого письма, благодаря чему он достаточно эффективен и хорошо сочетается с другими методами борьбы с спамом.

SMTP построен с учетом возможных сбоев доставки письма. В результате сбоя принимающего сервера письмо задерживается в очереди отправки на исходном почтовом сервере. Почтовый сервер периодически продолжает попытки доставки письма из очереди. Таким образом, единичный сбой в доставке письма приводит только к некоторой задержке доставки.

Однако, большое количество спама и вирусов рассылается специализированными программами, не имеющими очередей доставки. Такие программы, в отличие от штатных MTA, расчитаны на массовую доставку и просто игнорируют сбои доставки конкретным получателям.

Метод Greylisting основан этой на разнице в реализации SMTP-протокола в MTA и программах рассылки спама.

Ключевым понятием метода Greylisting является greylisting-триплет - комбинация из IP-адреса сервера отправителя письма, исходящего адреса SMTP-конверта и адреса получателя из SMTP-конверта.

Новое письмо, триплет которого ни разу не проходил в почтовом потоке сервера, отвергается с кодом временной ошибки (451). При следующей попытке доставки это письмо принимается, а его триплет заносится с базу данных SMTP-сервера. Последующие письма с таким же триплетом доставляются без задержек.

Таким образом, метод эффективно работает против программ рассылки спама и не оказывает влияния на установившийся почтовый поток сервера.

Использование метода Greylisting включается индивидуально для каждого списка доступа smtp-шлюза, благодаря чему можно выключать Greylisting для локальных сетей и «дружественных» внешних почтовых серверов.

Параметры работы Greylisting задаются в GUI разделе Конфигурация обработки почты в настройках smtp-шлюза.

Часто спам рассылается через открытые ретрансляторы — некорректно настроенные почтовые серверы, принимающие на ретрансляцию «чужие» сообщения. Такие серверы обычно существуют в сети небольшое время (до суток с момента обнаружения спамерами), после чего блокируются провайдерами и появляются в списках блокировки (ORBS и др). В отличие от специальных программ, такие серверы имеют очереди доставки и умеют повторно пересылать письма. Однако, метод Greylisting

способен блокировать и такой спам, хотя менее эффективно. Начальной задержки в 1 час часто достаточно для того, чтобы сервер появился в ORBS или был заблокирован провайдером. Кроме того, обычно очередь доставки открытого ретранслятора сильно перегружена, благодаря чему повторные попытки доставки происходят достаточно редко.

Примечание Для эффективной работы `Greylisting` должен быть включен на всех MX-серверах, принимающих почту для домена. Использование `Greylisting` только на главном сервере значительно снижает эффективность метода.

Для устранения ненужных задержек `Greylisting` должен быть выключен для писем, передающихся от между MX-серверами.

Примечание Некоторые менеджеры списков рассылки используют уникальный SMTP-конверт для каждого отправленного письма. Такие сообщения всегда будут задерживаться.

Дополнительную информацию о методе `Greylisting` можно получить на странице http://projects.puremagic.com/greylisting/whitepaper.html.

6.8 Прикладной шлюз протокола POP3

Прикладной шлюз протокола POP3 представляет собой отсоединенный процесс («демон») и предназначен для:

1. Фильтрации входящих и исходящих запросов на установление TCP-соединения по протоколу POP3 по сетевым адресам и портам источника и получателя запроса.
2. Отправки сообщений электронной почты по протоколу POP3.
3. Журналирования соединений.

Прикладной шлюз POP3 осуществляет передачу трафика протокола TCP в двух режимах — *прозрачном* и *непрозрачном*.

В *непрозрачном* режиме пользователь устанавливает TCP-соединение с МЭ по протоколу POP3, на котором выполняется прикладной шлюз POP3, на 110 порт. На основании заданных правил *политики безопасности* шлюз разрешает или отвергает установление соединения с пользователем. Если *политика безопасности* разрешает установление соединения данного пользователя с МЭ, шлюз переадресует TCP-соединение пользователя на удаленный сервер, определенный *политикой безопасности*.

Прозрачный режим работы универсального прикладного шлюза POP3 реализуется только совместно с пакетным фильтром (раздел 5.2.3 на стр. 57).

Режим аутентификации прикладным шлюзом POP3 не поддерживается.

6.8.1 Настройка прикладного шлюза POP3

Настройки прикладного шлюза состоят из:

- настроек резидентного процесса (раздел графического интерфейса `Настройки процесса`);
- настроек правил доступа (раздел графического интерфейса `Политика`).

6.8.1.1 Настройки резидентного процесса

Прикладной шлюз POP3 работает по `preforked` схеме, т.е. одновременно в системе запущено множество процессов, слушающих и обслуживающих запросы. Одновременно один процесс обслуживает одного клиента.

Настройки резидентного процесса подробно описаны в главе 6.1.2 на стр. 70. Для настроек прикладного шлюза POP3 служит следующий пункт в настройках:

`Устанавливать исходящий адрес клиента на интерфейсе` — имя интерфейса, на котором будет происходить подмена IP-адреса межсетевого экрана на IP-адрес клиента.

6.8.1.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к прикладному шлюзу протокола POP3 производится с помощью раздела графического интерфейса `Политика`. Отдельные правила отображаются графическим интерфейсом пользователя в виде подразделов. Правило доступа привязано к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Правило доступа прикладного шлюза POP3 состоит из:

- `Входящие правила` — правила контроля входящих соединений;
- `Исходящие правила` — правила контроля исходящих соединений;
- `Опции шлюза` — опции правила:

- `Хост назначения по умолчанию` — задает IP-адрес хоста, на который передаются TCP-соединения в «непрозрачном» режиме.
- `Порт назначения по умолчанию` — задает порт, на который передаются TCP-соединения в «непрозрачном» режиме.
- `Исходящий адрес` — указывает адрес, с которого будет осуществляться доступ с прикладного шлюза для данной группы.
  Т.е. можно определить, с какого именно IP-адреса МЭ (если их несколько) шлюз будет устанавливать соединения с внешними ресурсами для определенного списка доступа.

Фильтрация по IP-адресу и порту назначения указывается, как и у остальных прикладных шлюзов, в разделе `Исходящие правила` правил фильтрации.

6.9 Прикладной шлюз протокола FTP

Прикладной шлюз FTP предназначен для осуществления и контроля обмена файлами по протоколу FTP. Прикладной шлюз FTP осуществляет:

- ограничение доступа по IP-адресу и порту сервера;
- блокирование и протоколирования команд протокола FTP;
- поддержку прозрачного режима работы;
- поддержку безусловной переадресации запросов;
- аутентификацию пользователей на сервере аутентификации;
- передачу передаваемых файлов на антивирусную проверку (раздел 13 на стр. 165).

Прикладной шлюз FTP осуществляет обработку трафика протокола FTP в двух режимах: *прозрачном* и *непрозрачном*.

В непрозрачном режиме пользователь первоначально устанавливает FTP-соединение не с сервером, а с прикладным шлюзом FTP. Далее, используя специальную последовательность команд, пользователь производит соединение с необходимым ему сервером.

В прозрачном режиме все процедуры взаимодействия между FTP клиентом и прикладным шлюзом скрыты от пользователя. Для реализации прозрачного режима необходимы соответствующие настройки пакетного фильтра (раздел 5.2.3 на стр. 57).

Прикладной шлюз FTP поддерживает механизм безусловной переадресации. Он предназначен для доступа к FTP-серверам защищаемой информационной сети из общедоступной сети, а также для организации цепочек прикладных шлюзов FTP. При этом при получении FTP-запроса прикладной шлюз не интерпретирует указанный пользователем логин и устанавливает соединение с тем компьютером, где функционирует FTP-сервер. Его адрес и порт указываются администратором при настройке прикладного шлюза FTP. При этом фильтрация запросов на установление соединения не производится.

6.9.1 Настройка и конфигурация

Управление функционированием прикладного шлюза FTP осуществляется с помощью:

1. настроек резидентного процесса (раздел `Конфигурация резидентного процесса`);
2. настроек сервера аутентификации (раздел `Конфигурация сервера аутентификации`);
3. настроек правил доступа (раздел `Политика`).

6.9.1.1 Настройки резидентного процесса

Прикладной шлюз FTP работает по «forked» — схеме, т.е. изначально в системе запускается один процесс, ожидающий соединения по протоколу FTP. При приходе запроса создается новый процесс, которому передается запрос. Одновременно один процесс обслуживает одного клиента, при окончании FTP-сессии процесс завершается. Рекомендуемая величина максимального числа процессов — от 10 до 512.

Базовые настройки резидентного процесса описаны в главе 6.1.1 на стр. 70.

Прикладной шлюз FTP имеет следующие дополнительные настройки резидентного процесса:

`Имя файла с новостью дня` — полный путь к файлу, содержащему текст выводимого после установления соединения сообщения.

6.9.1.2 Правила доступа и фильтрации

Правила доступа привязаны к IP-адресу и порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Правило доступа прикладного шлюза FTP состоит из:

Входящие правила — правила контроля входящих соединений (раздел 6.2 на стр. 73).

Исходящие правила — правила контроля исходящих соединений. Функционирование происходит аналогично правилам проверки входящих соединений.

Опции шлюза — опции правила, состоящие из:

- **Включить авторизацию на шлюзе** — эта опция включает принудительную аутентификацию пользователей на прикладном шлюзе и более подробно описана в разделе 6.9.1.3 на стр. 101.
- **Родительский шлюз** — указывает IP-адрес «родительского» прикладного шлюза (раздел 6.9.1.4 на стр. 101).
- **Порт родительского шлюза** — указывает порт, на котором работает «родительский» шлюз.
- **Включить антивирусное сканирование** — задает необходимость антивирусной проверки передаваемых файлов.
- **URL сервера адаптации контента (ICAP)** — задает имя хоста антивирусного сервера.
- **Ограничение на количество соединений с одного IP** — определяет число максимального количества соединений с одного IP-адреса.

Запрещенные команды — список команд протокола FTP, использование которых запрещено. При получении одной из команд, перечисленных в этом списке, прикладной шлюз выдает пользователь сообщение о том, что данная команда заблокирована. В системный журнал записывается информация о заблокированной команде, передача поступившей команды не производится.

Журналируемые команды — список команд протокола FTP, при получении которых информация об этом заносится в системный журнал.

6.9.1.3 Аутентификация пользователей

Помимо фильтрации по IP-адресу/портам и командам FTP-протокола прикладной шлюз FTP поддерживает такое средство ограничения доступа как принудительная аутентификация.

Для включения режима принудительной аутентификации необходимо во вкладке `Опции шлюза` редактируемого ACL отметить галочкой пункт «Включить авторизацию на шлюзе». Кроме этого, в разделе `Сервер аутентификации` шлюза FTP следует указать IP-адрес и порт сервера аутентификации.

6.9.1.4 Безусловная переадресация в прикладном шлюзе FTP

Прикладной шлюз FTP поддерживает механизм безусловной переадресации. Для этого во вкладке `Опции шлюза` необходимо указать адрес и порт следующего в цепочке прикладного шлюза FTP. Адрес указывается в поле «Родительский шлюз», порт указывается в поле «Порт родительского шлюза». По умолчанию режим безусловной переадресации отключен и эти поля содержат значение «none».

Механизм работы безусловной переадресации следующий: при получении FTP-запроса прикладной шлюз FTP игнорирует адрес назначения, указанный пользователем, и устанавливает соединение с машиной, чьи адрес и порт указаны в полях «Родительский шлюз» и «Порт родительского шлюза». В режиме безусловной переадресации проверка на допустимость исходящего соединения к конечному адресу не производится (то есть игнорируются правила в разделе `Исходящие правила`).

Для того, чтобы отменить режим безусловной переадресации, необходимо в полях «Родительский шлюз» и «Порт родительского» шлюза установить значение «none».

Кроме взаимодействия с другими прикладными шлюзами, режим безусловной переадресации может быть использован для обеспечения доступа из сети Интернет к внутреннему FTP серверу, имеющему адрес из зарезервированного блока адресов. В качестве IP-адреса FTP-сервера этом случае указывается внешний адрес межсетевого экрана. Работа пользователей в режиме безусловной переадресации более подробно описана в разделе 6.9.2.3 на стр. 104.

6.9.2 Руководство пользователя по использованию прикладного шлюза FTP

В зависимости от настроек, заданных администратором МЭ, прикладной шлюз FTP может функционировать в различных режимах и в зависимости от этого механизм взаимодействия пользователя с прикладным шлюзом может различаться.

6.9.2.1 Работа в непрозрачном режиме

В непрозрачном режиме пользователь первоначально устанавливает FTP-соединение не с сервером, а с прикладным шлюзом FTP. Для соединения с необходимым сервером используется специальный формат имени пользователя: `username@server`. Получив данную команду, прикладной шлюз устанавливает соединение с указанным сервером, используя логин `username`. Пример работы через прикладной шлюз FTP приведен в разделе 6.9.2.1.1 на стр. 102.

В случае задания администратором авторизации через сервер авторизации пользователь осуществляет соединение с сервером в два приема. Первоначально производится авторизация на прикладном шлюзе FTP, а затем пользователь должен выдать вторую команду `user` для соединения с необходимым ему сервером. Пример работы через прикладной шлюз FTP с принудительной адресацией приведен в разделе 6.9.2.1.2 на стр. 103.

6.9.2.1.1 Пример работы шлюза в непрозрачном режиме

В этом случае пользователь устанавливает FTP-соединение на межсетевой экран, после чего попадает в диалог ввода имени пользователя:

```
asv@snowball:~[133]$ ftp next
Connected to next.service.jet.msk.su.
220-[ Warning ! ]
220-[ Unauthorized access to this computer system is prohibited.  ]
220-[ All connections are logged. ]
220 FTP proxy ready to service. Name (snowball:asv):
```

Далее пользователь должен ввести имя пользователя и адрес или имя хоста, с которым пользователь хочет установить соединение. Формат этого запроса следующий:

```
<имя пользователя>@<имя сервера>
```

Например, если пользователь хочет получить соединение с FTP-сервером python от имени пользователя asv, то он должен ввести строчку asv@python:

```
Name (snowball:asv): asv@python
```

После этого прикладной шлюз FTP устанавливает соединение с требуемым сервером:

```
331-[ 220 You are connected to 192.168.10.6 ]
331-[ 220 python FTP server (UNIX(r) System V Release 4.0) ready. ]
331 Password required for asv.
```

```
Password:
230 User asv logged in.
ftp>
```

6.9.2.1.2 Пример работы шлюза в непрозрачном режиме с принудительной аутентификацией

В этом случае пользователь сначала должен аутентифицироваться на прикладном шлюзе:

```
asv@snowball:~[53]$ ftp next
Connected to next.service.jet.msk.su.
220-[ Warning ! ]
220-[ Unauthorized access to this computer system is prohibited.  ]
220-[ All connections are logged. ]
220-FTP proxy ready to service.
220 You have to authenticate on proxy first
Name (next:asv):test1
331 Enter password for user test1
Password:
230 User authenticated to proxy.
ftp>
```

После этого пользователь инициирует соединение с нужным ему FTP-сервером. Формат этого запроса следующий:

```
user <имя пользователя>@<имя сервера>
```

Например, если пользователь хочет установить соединение с FTP-сервером python от имени пользователя asv, то он должен ввести строчку user asv@python:

```
ftp> user asv@python
```

После этого прикладной шлюз FTP устанавливает соединение с требуемым сервером:

```
331-[ 220 You are connected to 192.168.10.6 ]
331-[ 220 python FTP server (UNIX(r) System V Release 4.0) ready. ]
331 Password required for asv.
Password:
230 User asv logged in.
ftp>
```

6.9.2.2 Работа шлюза в «прозрачном» режиме

При работе в «прозрачном» режиме пользователь общается с FTP-сервером как обычно, и непосредственного диалога между прикладным шлюзом и пользователем не происходит.

Исключением является работа прикладного шлюза FTP с принудительной аутентификацией пользователей через сервер аутентификации. В этом случае после аутентификации на шлюзе пользователь должен ввести команду `user имя_пользователя`, а затем выдать вторую команду `user` для аутентификации на запрашиваемом FTP-сервере.

6.9.2.2.1 Пример работы шлюза в «прозрачном» режиме с принудительной аутентификацией

В этом случае пользователь сначала аутентифицируется на прикладном шлюзе:

```
asv@snowball:~[53]$ ftp python
Connected to python.service.jet.msk.su.
220-[ Warning ! ]
220-[ Unauthorized access to this computer system is prohibited.  ]
220-[ All connections are logged. ]
220-FTP proxy ready to service.
220 You have to authenticate on proxy first
Name (next:asv):test1
331 Enter password for user test1
Password:
230 User authenticated to proxy.
ftp>
```

После этого пользователь вводит идентификатор, от имени которого будет установлено соединение с FTP-сервером. Формат этого запроса следующий:

```
user <имя пользователя>
```

Например, если пользователь хочет зарегистрироваться на FTP-сервере пользователем с идентификатором asv, то он вводит строчку «user asv»:

```
ftp> user asv
```

После этого прикладной шлюз FTP устанавливает соединение с требуемым сервером:

```
331-[ 220 You are connected to 192.168.10.6 ]
331-[ 220 python FTP server (UNIX(r) System V Release 4.0) ready. ]
331 Password required for asv.
Password:
230 User asv logged in.
ftp>
```

6.9.2.3 Работа шлюза в режиме с безусловной переадресацией

При работе в режиме с безусловной переадресацией пользователь общается с FTP-сервером как обычно. Например, если прикладной шлюз FTP настроен на безусловную переадресацию FTP-соединений на сервер python, сеанс работы через прикладной шлюз будет выглядеть следующим образом:

```
asv@snowball:~[53]$ftp next
Connected to next.
220-[ Warning ! ]
220-[ Unauthorized access to this computer system is prohibited.  ]
220-[ All connections are logged. ]
220 FTP proxy [version 1.5.3] ready to service.
Name (snowball:asv): asv
  331-[ 220 You are connected to 192.168.10.6 ]
331-[ 220 python FTP server (UNIX(r) System V Release 4.0) ready. ]
331 Password required for asv.
Password:
230 User asv logged in.
ftp>
```

6.10 Универсальный прикладной шлюз протокола UDP

6.10.1 Алгоритм функционирования

UDP представляет собой протокол транспортного уровня, предусматривающий передачу по сети отдельных пакетов данных — *UDP-дейтаграмм*. Протокол UDP не устанавливает соединение и не поддерживает сессии. Проверка того, дошла ли информация до адресата, также не производится. Прикладной шлюз протокола UDP предназначен для контроля прохождения UDP-дейтаграмм через МЭ «Z-2».

Использование прикладного шлюза протокола UDP позволяет:

1. Разрешать или запрещать прохождение UDP-дейтаграммы на основе анализа значений IP-адресов и портов отправителя и получателя. Таким образом, МЭ «Z-2» осуществляет управление обменом по протоколу UDP.
2. Поддерживать *псевдосессии* вида запрос-ответ .
3. Протоколировать прохождение UDP-дейтаграмм через межсетевой экран.

Универсальный прикладной шлюз UDP является нейтральным по отношению к содержимому протокола и может быть использован в качестве туннеля для любого протокола, использующего в качестве транспорта протокол UDP.

Универсальный прикладной шлюз UDP позволяет производить передачу UDP-дейтаграмм в двух режимах: *прозрачном* и *непрозрачном*.

В *непрозрачном* режиме все сервисы, использующие протокол UDP, должны быть перенастроены так, чтобы обращения всегда производились на тот компьютер, где установлен универсальный прикладной шлюз UDP. Получив дейтаграмму, прикладной шлюз отправляет ее на тот компьютер, где функционирует требуемый сервис. Таким образом, на каждый UDP-сервис запускается отдельный шлюз UDP. Использование непрозрачного режима требует явного указания прикладному шлюзу IP-адреса/порта назначения, куда должен быть передан UDP-пакет.

Прозрачный режим позволяет отказаться от необходимости перенастраивать все службы, которые используют UDP-протокол. В этом случае перенаправление пакетов выполняется средствами пакетного фильтра. Значения IP-адреса и порта назначения считываются из заголовка исходной UDP-дейтаграммы. Прозрачный режим работы универсального прикладного шлюза UDP реализуется только совместно с пакетным фильтром (раздел 5.2.3 на стр. 57).

Режим аутентификации с использованием сервера аутентификации универсальный прикладной шлюз UDP не поддерживает.

В обоих режимах в качестве IP-адреса отправителя ретранслируемых пакетов используется IP-адрес компьютера, где функционирует ПО межсетевого экрана, а порт выбирается произвольным. Далее шлюз ожидает ответ удаленного сервиса на этом выбранном порту и производит передачу полученного ответа. В течение заданного времени шлюз UDP сохраняет таблицу соответствия между IP-адресом/портом клиента, пославшего запрос, с IP-адресом и портом МЭ, с которого был ретранслирован этот запрос на удаленный сервис. Такая таблица создает *псевдосессии* для обеспечения корректного двустороннее прохождение UDP-дейтаграмм через универсальный прикладной шлюз UDP.

На рис. 6.7 показана схема прохождения UDP-дейтаграммы без использования прикладного шлюза, а на рис. 6.8 с использованием прикладного шлюза.

6.10.2 Настройка универсального шлюза UDP

Настройки шлюза UDP состоят из:

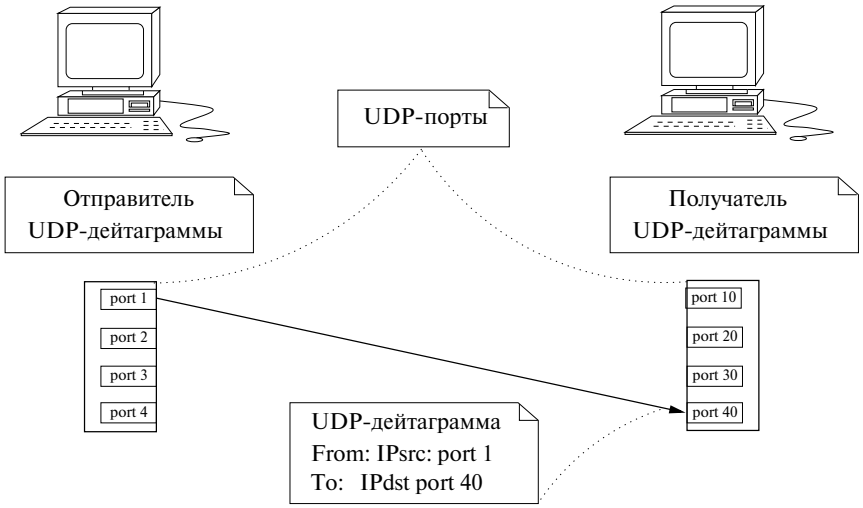

Рис. 6.7: Прохождение UDP–дейтаграммы в отсутствие прикладного шлюза UDP.

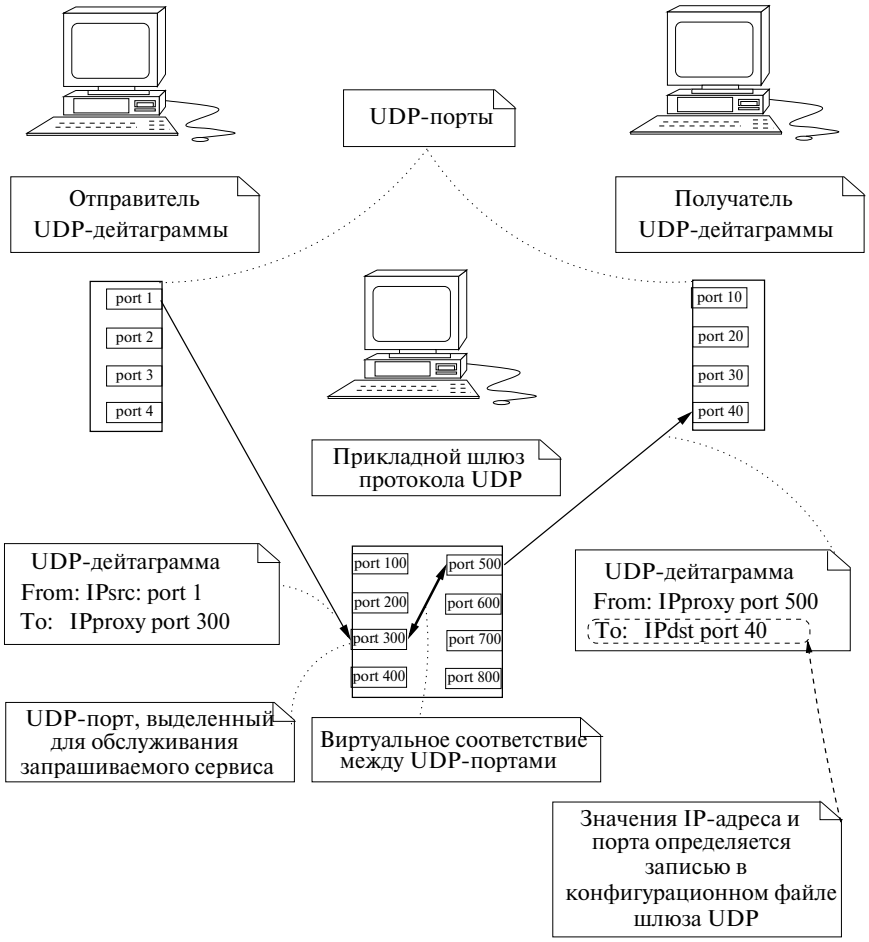

Рис. 6.8: Универсальный прикладной шлюз протокола UDP. Прохождение UDP–дейтаграммы.

- настроек резидентного процесса (раздел графического интерфейса `Настройки процесса`);
- настроек правил доступа (раздел графического интерфейса`Политика`).

6.10.2.1 Настройки резидентного процесса

Универсальный прикладной шлюз UDP ретранслирует все пакеты в рамках одного процесса.

Настройки резидентного процесса осуществляются в разделе `Настройки процесса` и подробно описаны в главе 6.1.3 на стр. 71.

6.10.2.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к универсальному прикладному шлюзу протокола UDP производится с помощью раздела `Политика`. Отдельные правила отображаются графическим интерфейсом пользователя в виде подразделов. Правило доступа привязано к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Правило доступа универсального прикладного шлюза UDP состоит из:

- `Входящие правила` — правила контроля входящих соединений
- `Исходящие правила` — правила контроля исходящих соединений
- `Опции шлюза` — опции правила:
  - `Хост назначения по умолчанию` — задает хост, на который передаются датаграммы в «непрозрачном» режиме.
  - `Порт назначения по умолчанию` — задает порт, на который передаются пакеты в «непрозрачном» режиме.
  - `Исходящий адрес` — указывает адрес, с которого будет осуществляться доступ с прикладного шлюза для данной группы.
    Т.е. можно определить, с какого именно IP-адреса МЭ (если их несколько) шлюз будет устанавливать соединения с внешними ресурсами для определенного списка доступа.

Фильтрация по IP-адресу и порту назначения указывается, как и у остальных прикладных шлюзов, в разделе `Исходящие правила` правил фильтрации.

6.11 Прикладной шлюз протокола TCP

Универсальный шлюз протокола TCP представляет собой отсоединенный процесс («демон») и предназначен для:

1. Фильтрации входящих и исходящих запросов на установление TCP-соединения по сетевым адресам и портам источника и получателя запроса.
2. Журналирования соединений.

Универсальный прикладной шлюз TCP является нейтральным по отношению к содержимому протокола и может быть использован в качестве туннеля для любого протокола, использующего в качестве транспорта протокол TCP.

Прикладной шлюз TCP осуществляет передачу трафика протокола TCP в двух режимах — *прозрачном* и *непрозрачном*.

В *непрозрачном* режиме пользователь устанавливает TCP-соединение с МЭ, на котором выполняется универсальный прикладной шлюз TCP на заранее известный порт. На основании заданных правил (*политики безопасности*) шлюз разрешает или отвергает установление соединения с пользователем. Если *политика безопасности* разрешает установление соединения данного пользователя с МЭ, шлюз переадресует TCP-соединение пользователя на удаленный сервер, определенный *политикой безопасности*.

Прозрачный режим работы универсального прикладного шлюза TCP реализуется только совместно с пакетным фильтром (раздел 5.2.3 на стр. 57).

Режим аутентификации универсальным прикладным шлюзом TCP не поддерживается.

6.11.1 Настройка универсального прикладного шлюза TCP

Настройки прикладного шлюза состоят из:

- настроек резидентного процесса (раздел графического интерфейса `Настройки процесса`);
- настроек правил доступа (раздел графического интерфейса `Политика`).

6.11.1.1 Настройки резидентного процесса

Универсальный прикладной шлюз TCP работает по `preforked` схеме, т.е. одновременно в системе запущено множество процессов, слушающих и обслуживающих TCP-запросы. Одновременно один процесс обслуживает одного клиента.

Настройки резидентного процесса подробно описаны в главе 6.1.2 на стр. 70. Для настроек универсального прикладного шлюза добавлен следующий пункт в настройках:

`Устанавливать исходящий адрес клиента на интерфейсе` — имя интерфейса, на котором будет происходить подмена IP-адреса межсетевого экрана на IP-адрес клиента.

6.11.1.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к универсальному прикладному шлюзу протокола TCP производится с помощью раздела графического интерфейса `Политика`. Отдельные правила отображаются графическим интерфейсом пользователя в виде подразделов. Правило доступа привязано к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Правило доступа универсального прикладного шлюза TCP состоит из:

- `Входящие правила` — правила контроля входящих соединений;

- `Исходящие правила` — правила контроля исходящих соединений;
- `Опции шлюза` — опции правила:
  - `Порт назначения по умолчанию` — задает IP-адрес хоста, на который передаются TCP-соединения в «непрозрачном» режиме;
  - `Порт назначения по умолчанию` — задает порт, на который передаются TCP-соединения в «непрозрачном» режиме.
  - `Исходящий адрес` — указывает адрес, с которого будет осуществляться доступ с прикладного шлюза для данной группы.
    Т.е. можно определить, с какого именно IP-адреса МЭ (если их несколько) шлюз будет устанавливать соединения с внешними ресурсами для определенного списка доступа.

Фильтрация по IP-адресу и порту назначения указывается, как и у остальных прикладных шлюзов, в разделе `Исходящие правила` правил фильтрации.

6.11.2 Примеры конфигурации универсального прикладного шлюза TCP

Предположим, необходимо обеспечить доступ к серверу www.my-domain.ru из сети Интернет по протоколу HTTP. Это можно сделать, используя либо пакетный фильтр, либо с помощью универсального прикладного шлюза TCP. В данном примере мы будем рассматривать решение с использованием универсального шлюза TCP.

Для этого необходимо создать экземпляр этого шлюза (назовем его www-gw), принимающего соединения на порту 80 внешнего интерфейса МЭ (рисунок 6.9).

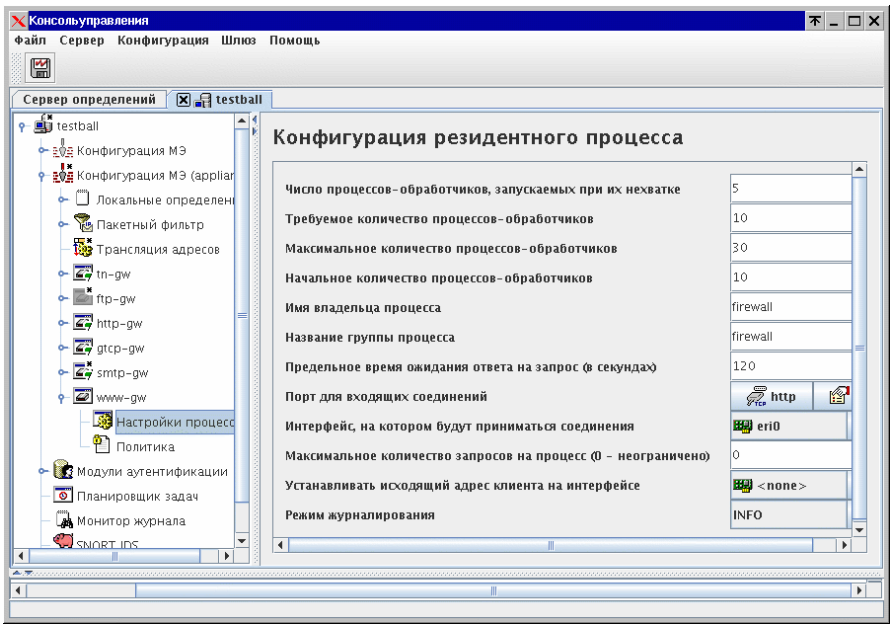

Рис. 6.9: Настройка шлюза `www-gw`

В настройках ACL в разделе `Входящие правила` необходимо разрешить доступ со всех адресов, а в разделе `Опции прокси` указать сервер www.my-domain.ru и порт 80 в качестве значений пунктов `Хост назначения по умолчанию` и `Порт назначения по умолчанию` (рисунок 6.10). В качестве значения пункта `Исходящий адрес` указывается адрес внешнего интерфейса межсетевого экрана.

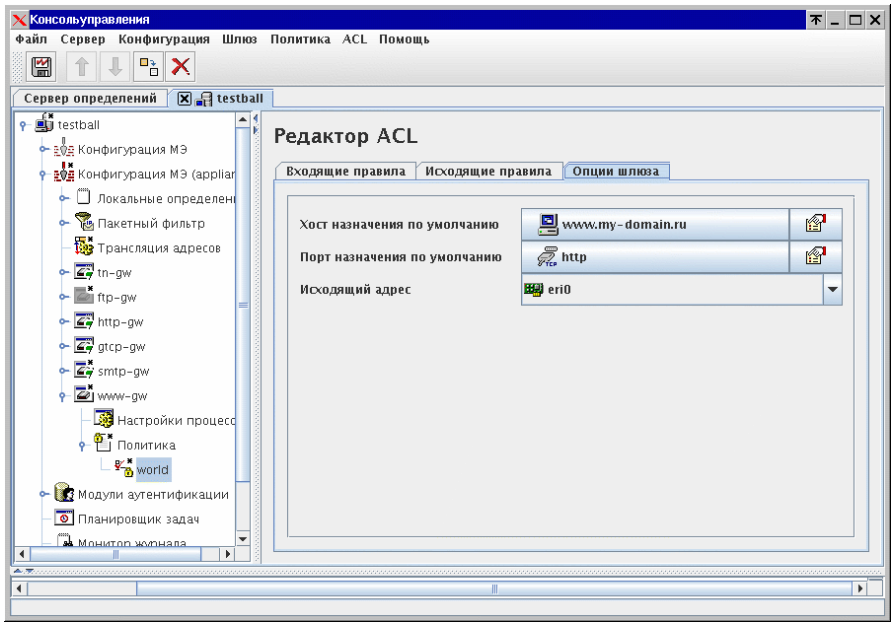

Рис. 6.10: Настройка шлюза `www-gw`

При такой конфигурации все соединения, адресованные на внешний интерфейс МЭ на порт 80, будут перенаправлены на HTTP сервер www.my-domain.ru. Если мы хотим, чтобы к нашему HTTP-серверу пользователи сети Интернет обращались не как к внешнему адресу МЭ (208.172.16.2), а как к отдельному адресу (208.172.16.2 в нашем примере), необходимо модифицировать ARP-таблицу на межсетевом экране, как это описано в разделе 5.2.5 на стр. 59.

В вышеописанной конфигурации МЭ существует небольшой недостаток: так как все соединения к корпоративному HTTP-серверу www.my-domain.ru устанавливаются от имени прикладного шлюза `www-gw`, администратор данного сервера лишается статистики посещения своего сервера — ведь единственным адресом, с которого устанавливаются соединения к серверу, является адрес интерфейса iprb0 межсетевого экрана.

Для решения подобной проблемы в универсальном прикладном шлюзе протокола TCP существует опция `Устанавливать исходящий адрес клиента на интерфейсе`. В нашем примере необходимо в качестве значения этой опции установить адрес внутреннего интерфейса межсетевого экрана (iprb1). В этом случае все соединения, создаваемые прикладным шлюзом `www-gw`, при выходе с интерфейса iprb1 будут иметь оригинальный адрес инициатора соединения. При таком режиме работы обеспечивается безопасный доступ через прикладной шлюз и решается проблема с сокрытием оригинальных адресов прикладным шлюзом. Следует заметить, что при работе `www-gw` в этом режиме необходимо чтобы:

- сервер www.my-domain.ru имел правило маршрутизации по умолчанию, указывающее на адрес интерфейса iprb1 межсетевого экрана;
- в настройках трансляции адресов МЭ присутствовало хотя бы одно правило.

Более подробно режим сохранения исходного адреса описан в разделе 6.4 на стр. 78.

6.12 Прикладной шлюз протокола telnet

6.12.1 Алгоритм функционирования

Прикладной шлюз протокола telnet обеспечивает возможность прохождения telnet-сессий через МЭ «Z-2». Использование шлюза telnet позволяет:

1. разрешать или запрещать установление telnet-соединения на основе анализа значений IP-адресов и портов отправителя и получателя запроса;
2. аутентифицировать пользователей шлюза с использованием сервера аутентификации;
3. протоколировать установление соединений;
4. изменять пользователю собственный пароль;
5. использовать «прозрачный» режим.

После установления соединения прикладной шлюз telnet не выполняет контроль содержимого передаваемых данных.

6.12.2 Настройки прикладного шлюза telnet

Настройки прикладного шлюза telnet состоят из:

- настроек резидентного процесса (раздел графического интерфейса `Настройки процесса`);
- настроек сервера аутентификации (раздел графического интерфейса `Сервер аутентификации`);
- настроек правил доступа (раздел графического интерфейса `Политика`).

Функционирование шлюза telnet возможно в двух режимах: *непрозрачном прозрачном*. В *непрозрачном* режиме пользователь устанавливает соединение с помощью утилиты telnet на тот компьютер, на котором запущен шлюз telnet, используя заранее известный порт и в диалоговом режиме задает необходимый ему сервер назначения. Подробно диалог описан в разделе 6.12.3 на стр. 113.

Прозрачный режим работы прикладного шлюза telnet реализуется только совместно с пакетным фильтром. Для реализации *прозрачного* режима пакетный фильтр должен быть соответствующим образом настроен (раздел 5.2.3 на стр. 57).

В *прозрачном* режиме пользователь, устанавливающий соединение с удаленным сервером, не замечает присутствия шлюза.

6.12.2.1 Настройки резидентного процесса

Прикладной шлюз telnet работает по «forked»-схеме, т.е. изначально в системе запускается один процесс, ожидающий соединения пользователей. При поступлении очередного запроса на установление соединения, создается новый процесс, который обслуживает соединение. Одновременно один процесс обслуживает одного клиента, при окончании сессии этот процесс завершает свою работу.

Рекомендуемая величина максимального числа процессов — от 10 до 512.

Настройки резидентного процесса описаны в главе 6.1.1 на стр. 70. Для настройки шлюза telnet добавлены следующие пункты настроек:

`Устанавливать исходный адрес клиента на интерфейсе` — имя интерфейса, на котором будет происходить подмена IP-адреса межсетевого экрана на IP-адрес клиента;

`Имя файла с новостью дня` — полный путь к файлу, содержащему текст выводимого после установления соединения сообщения;

`Стока приглашения` — строка приглашения telnet. Выводится в ходе диалога пользователя с шлюзом.

6.12.2.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к прикладному шлюзу протокола telnet производится с помощью раздела `Политика`. Отдельные правила отображаются графическим интерфейсом пользователя в виде подразделов. Правило доступа привязано к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Правило доступа для прикладного шлюза telnet состоит из:

`Входящие правила` — правила контроля входящих соединений;
`Исходящие правила` — правила контроля исходящих соединений;
`Опции прокси` — опции правила:

- `Включить авторизацию на прикладном шлюзе` — включает аутентификацию пользователей на прикладном шлюзе, см. раздел 6.12.2.3 на стр. 113.
- `Хост родительского прикладного шлюза` — указывает родительский telnet-шлюз;
- `Порт родительского прикладного шлюза` — указывает порт, на котором слушает «родительский» telnet-шлюз (раздел 6.6.5 на стр. 84);
- `Исходящий адрес` — указывает адрес, с которого будет осуществляться доступ с прикладного шлюза для данной группы.
  Т.е. можно определить, с какого именно IP-адреса МЭ (если их несколько) шлюз будет устанавливать соединения с внешними ресурсами для определенного списка доступа.
- `Ограничение на количество соединений с одного IP` — определяет число максимального количества соединений с одного IP-адреса.

6.12.2.3 Аутентификация пользователей

Прикладной шлюз telnet поддерживает аутентификацию пользователей через сервер аутентификации.

Настройка IP-адреса и порта сервера аутентификации осуществляется в разделе `Сервер аутентификации`.

Кроме аутентификации, шлюз telnet предоставляет возможность пользователям менять их собственные пароли.

6.12.3 Диалог пользователя с прикладным шлюзом

В *непрозрачном* режиме пользователь для установления соединения с сервером назначения ведет диалог с прикладным шлюзом telnet.

Прикладной шлюз telnet поддерживает следующие диалоговые команды:

Таблица 6.2: **Команды диалога с пользователем прикладного шлюза telnet**

Имя команды	Параметры	Комментарий
connect	hostname [serv/port]	Команда на установление соединения по протоколу telnet с указанным параметром *hostname* удаленным сервером на порт, заданный параметром *[serv/port]*. Параметр *hostname* может представлять собой IP-адрес или имя удаленного сервера. Параметр *[serv/port]* задает номер порта или имя сервиса на удаленном сервере, с которым должно быть установлено соединение. Имена сервисов разрешаются посредством обращения к файлу /etc/services. Параметр *[serv/port]* является необязательным.

Продолжение следует

. . . продолжение таблицы 6.2.

telnet	см. выше.	Синоним команды *connect*.
passwd	нет	Позволяет пользователю сменить текущий пароль. Использование данной команды возможно только в после аутентификации на прикладном шлюзе протокола telnet. Если возможность смены пароля для данного пользователя заблокирована в настройках сервера аутентификации, то все попытки смены пароля будут оканчиваться неудачей.
help	нет	Команда *help* приводит к выдаче пользователю краткой справки о командах пользовательского диалога.
?	нет	Синоним команды *help*
quit	нет	Команда приводит к завершению сеанса работы пользователя со шлюзом telnet и закрытию соединения. Синонимом команды *quit* является команда *exit*. Кроме того, к завершению сеанса приводит и поступивший от пользователя кодовый символ ^D.
exit	нет	Синоним команды *quit*

6.12.4 Пример конфигурации

Предположим, что следует обеспечить маршрутизацию соединений протоколом telnet к корпоративному серверу приложений app-server.my-domain.ru. Прикладная программа (предположим, что некоторая разновидность системы «клиент-банк») ожидает соединений на порт 23/TCP.

Клиентами программы являются как внешние пользователи (из сети интернет), так и пользователи корпоративной сети. Доступ к к корпоративному серверу приложений должен осуществляться в прозрачном режиме с аутентификацией.

Доступ протоколом telnet из внутренней сети во внешние сети (кроме сети *DMZ*) через МЭ «Z-2» разрешен в непрозрачном режиме.

Итак, следует обеспечить маршрутизацию соединений протоколом telnet через МЭ «Z-2» по следующей схеме:

1. разрешен доступ к компьютеру app-server.my-domain.ru протоколом telnet на порт 23/TCP в прозрачном режиме с аутентификацией;
2. разрешен доступ протоколом telnet из внутренней сети (LAN) во внешние сети (Интернет);
3. запрещен доступ из сети *DMZ* во внутреннюю сеть;
4. запрещен доступ из внешних сетей во внутреннюю сеть и ко всем компьютерам сети *DMZ* кроме app-server.my-domain.ru.

Данная задача может быть решена средствами МЭ «Z-2» следующим образом:

- все исходящие соединения из сети *DMZ* блокируются средствами пакетного фильтра. Причем, это требование касается не только telnet-соединений;
- для обеспечения доступа к серверу приложений на МЭ активируется прикладной шлюз tn-gw-app, ожидающий входящие соединения на всех интерфейсах на порт 23 и выполняющий безусловную переадресацию соединения на порт 23 компьютера app-server.my-domain.ru (см. рис. 6.11). В правилах доступа для прикладного шлюза tn-gw-app указывается, что пользователям из всех сетей разрешено устанавливать соединения только на адрес app-server.my-domain.ru;
- для обеспечения прозрачного режима функционирования прикладного шлюза средствами пакетного фильтра осуществляется переадресация соединений, устанавливающихся напрямую на адрес app-server.my-domain.ru (208.172.16.5) на прикладной шлюз;
- для обеспечения доступа пользователей из внутренней сети во внешние сети, активируется прикладной шлюз tn-gw-ext, ожидающий входящие на интерфейсе iprb0, порт 2323 (см. рис. 6.12). В правилах доступа для прикладного шлюза tn-gw-ext указывается, что разрешены исходящие

соединения во внешние сети для пользователей сети LAN. Доступ в сеть *DMZ* через прикладной шлюз tn-gw-ext запрещен (см. рис. 6.13). Прикладной шлюз tn-gw-ext функционирует в непрозрачном режиме.

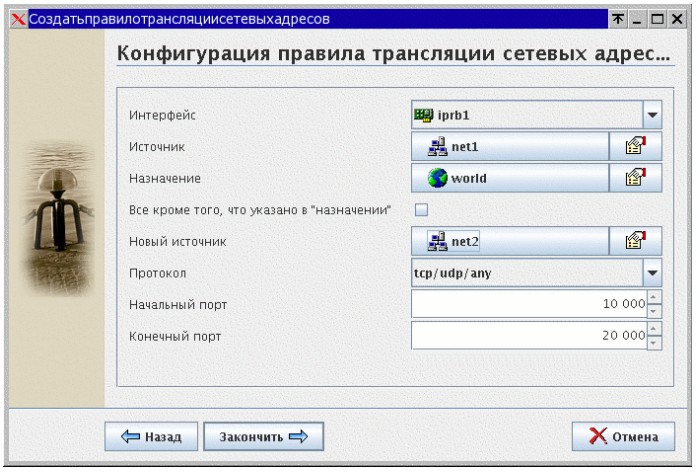

Рис. 6.11: Настройка адресата назначения для шлюза tn–gw–app

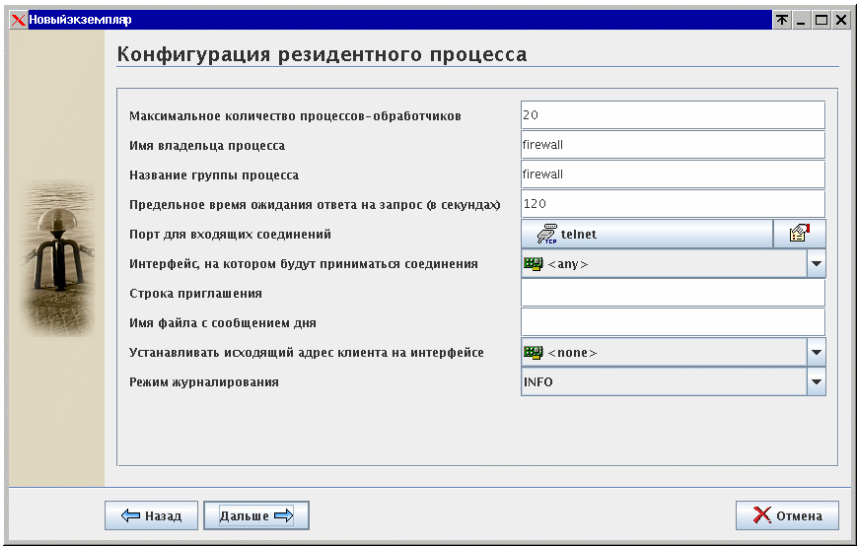

Рис. 6.12: Настройка прикладного шлюза tn–gw–ext

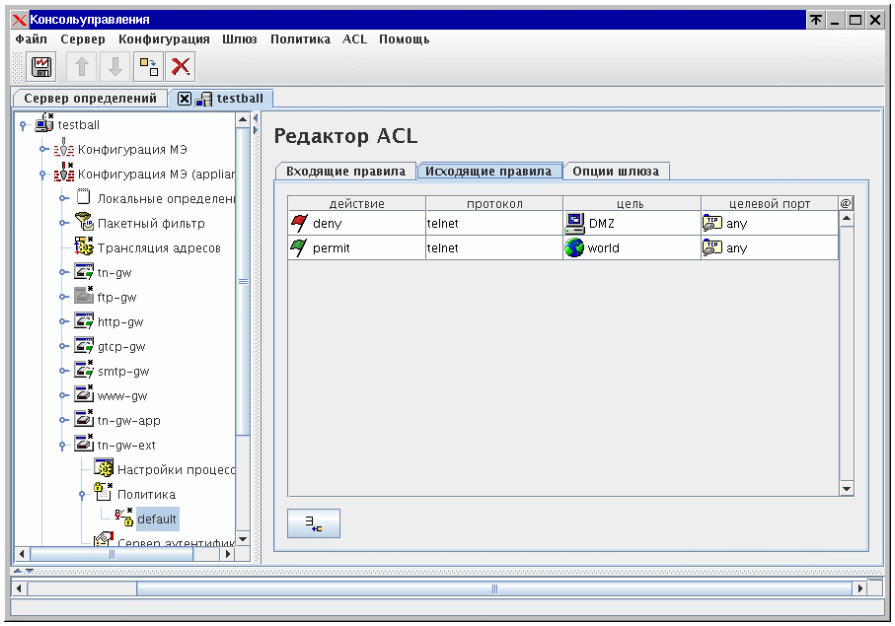

Рис. 6.13: Настройка правил доступа для прикладного шлюза tn–gw–ext

6.13 Прикладной шлюз протокола SNMP

6.13.1 Краткие сведения о протоколе SNMP

Протокол SNMP (`Simple Network Management Protocol` или «простой протокол управления сетью») предназначен для мониторинга сетевой активности и управления сетью.

На сегодняшний день разработано несколько версий протокола SNMP. Практический интерес представляют три:

SNMPv1 — первая версия протокола SNMP, имеющая статус стандарта (rfc 1155-1157); является наиболее распространенной среди поддерживаемых версий SNMP среди производителей сетевого оборудования; предполагает обмен пакетами только протоколом UDP;

SNMPv2c (Community-based SNMPv2) — вторая версия протокола SNMP, получившая распространение, имеет статус стандарта (rfc 1901-1908); поддерживается некоторыми производителями сетевого оборудования и ПО управления сетью; позволяет осуществлять обмен SNMP-пакетами как с использованием UDP, так и TCP;

SNMPv3 — третья версия протокола SNMP (rfc 2571-2575); пока не имеет статус стандарта и практически не поддерживается производителями сетевого оборудования; позволяет осуществлять обмен SNMP-пакетами как с использованием UDP, так и TCP, обеспечивает защиту SNMP-трафика.

Примечание Прикладной шлюз протокола SNMP поддерживает протоколы SNMPv1 и SNMPv2c и UDP-транспорт.

В дальнейшем будем рассматривать протоколы `SNMPv1` и `SNMPv2c`. Если ниже в тексте встречается фраза «протокол SNMP» без указания версии, то сказанное относится к обеим рассматриваемым версиям.

Протокол SNMP работает поверх транспортного протокола UDP и предназначен для использования сетевыми управляющими станциями.

На транспортном уровне протокол представляет собой простой обмен сообщениями конечной длины без сохранения состояния. Протокол реализует как модель «запрос-ответ», так и посылку уведомлений о произошедших событиях.

Обмен SNMP-пакетами происходит между управляющей программой (`SNMP-менеджером`) и объектом управления (`SNMP-агента`). Обычно SNMP-агент реализуется в виде отдельного процесса на управляемом компьютере или является встроенной функцией сетевого оборудования. SNMP-менеджер посылает запросы SNMP-агенту и принимает от него ответы на запросы. SNMP-агент может по собственной инициативе посылать уведомления о произошедших событиях(`SNMP Traps`), которые не требуют подтверждения.

На логическом уровне, с точки зрения протокола SNMP, объект управления представлен некоторым множеством переменных, имеющих тип и значение. Протокол позволяет прочитать и/или изменить значение некоторой переменной. Протокол не специфицирует «семантику» значений переменных. Например, «присваивание» переменной «ShutdownTime» значения 60 может вызвать перезагрузку управляемого компьютера через 60 секунд.

Переменные, значения которых изменяются протоколом, идентифицируются по номеру и по имени и организованы в единую базу данных, которая называется `MIB` — management information base.

Информация, содержащаяся в `MIB` может описывать неограниченное количество переменных управления. Каждая переменная имеет уникальное имя (`object identifier`, `OID`). Передаваемая по сети информации всегда представляет собой одну или несколько пар вида `OID=значение`. Рассмотренные ниже команды позволяют менеджеру запросить значения некоторого подмножества `OID` или, наоборот, установить значения одного или нескольких объектов. Интерпретацию значений переменных и возможные действия осуществляет `SNMP`-агент.

Все существующие в мире OID организованы в одну большую древовидную структуру. Последовательности чисел, которые представляют собой OID — это идентификаторы ветвей дерева. Каждое поддерево в дереве назначается централизованно, что гарантирует уникальность OID. Например, ветвь дерева MIB, зарегистрированная на компанию «Инфосистемы Джет» имеет номер 1.3.6.1.4.1.11821 и соответствует имени iso.org.dod.internet.private.enterprise.11821.

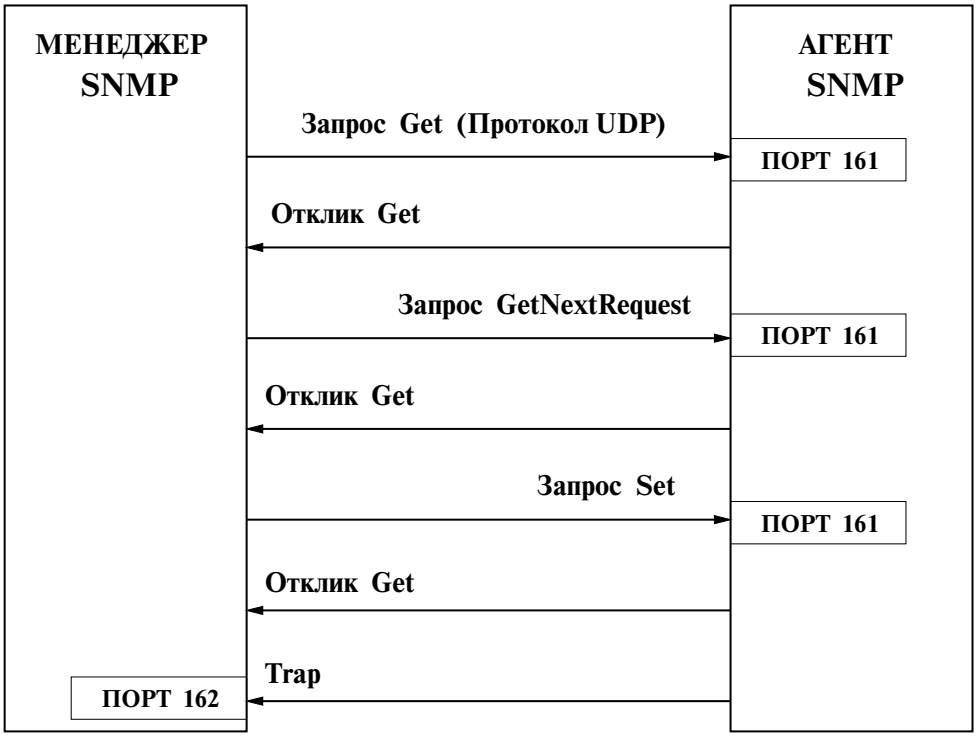

Рис. 6.14: Схема запросов/ответов SNMP

Взаимодействие между агентом и менеджером осуществляется при помощи команд. Схема взаимодействия приведена на рис. 6.14. При этом используются два порта для двустороннего общения: порт для обычных сообщений (по умолчанию 161) и порт для информирования о происходящих событиях сетевых объектов — snmp-trap (порт 162). Состав команд различается в первой и второй версии протокола SNMP. Их состав приведен в таблицах 6.3 и 6.4.

Примечание Третью версию протокола SNMP ПО МЭ «Z-2» не поддерживает.

Таблица 6.3: **Команды протокола snmp. Первая версия**

Команда SNMP	Назначение
GetRequest	Получить значение указанной переменной или информацию о состоянии сетевого элемента.
GetNextRequest	Получить значение переменной, не зная ее точного ее имени, т.е. получить следующий логический идентификатор на дереве MIB.
SetRequest	Присвоить переменной соответствующее значение. Используется для описания действия, которое должно быть выполнено.
GetResponse	Отклик на GetRequest, GetNextRequest, SetRequest. Содержит также информацию о состоянии (коды ошибок и другие данные).
Trap	Сообщение сетевого объекта о происшедшем событии.

Вторая версия протокола snmp (SNMPv.2) определяет три новые команды:

Таблица 6.4: **Дополнительные команды второй версии протокола snmp.**

Команда SNMP	Назначение
InformRequest	Позволяет одному менеджеру послать сообщение типа Trap другому менеджеру и запросить ответ
GetBulkRequest	Позволяет менеджеру получать большие блоки данных с максимальной эффективностью.
Report	Неспецифицированная команда. Формат определяется конкретной реализацией.

Вторая версия протокола SNMP поддерживает все команды, определенные в первой версии. Кроме того, во второй версии определены команды TRAP2 и RESPONSE, эквивалентные соответственно командам TRAP и GETRESPONSE первой версии.

SNMP-агенты всегда объединяются в сообщества, в рамках которых устанавливаются одинаковые права доступа. SNMP-запрос содержит последовательность символов, которая является идентификатором сообщества, к которому относится запрашиваемый агент.

Кроме того, в SNMP-запросах используется уникальное число — идентификатор. Его значение устанавливается менеджером и возвращается агентом в процессе ответа. Это позволяет связывать в пары запросы и ответы на них.

6.13.2 Описание прикладного шлюза протокола SNMP

Прикладной шлюз протокола SNMP позволяет:

1. поддерживать режим обмена с «сохранением состояния» (контролировать прохождения ответов на запросы);
2. осуществлять фильтрацию запросов, отправляемых SNMP-менеджерами к SNMP-агентам. В качестве критериев фильтрации используются:
   (а) адреса и порты источника и назначения;
   (б) номер версии SNMP-протокола;
   (в) указанное в запросе сообщество (community) SNMP-агентов должно находится в списке разрешенных;
   (г) команды SNMP-протокола.

Сообщения, не отвечающие хотя бы одному критерию, удаляются с занесением соответствующего уведомления в протокол. Отправитель SNMP-сообщения никаких уведомлений при этом не получает.

Режим аутентификации с использованием сервера аутентификации прикладной шлюз SNNP не поддерживает.

6.13.3 Настройки прикладного шлюза SNMP

Настройки прикладного шлюза SNMP состоят из:

- настроек резидентного процесса (раздел Настройки процесса);
- настроек правил доступа (раздел Политика).

Прикладной шлюз SNMP поддерживает два режима работы: *прозрачный* и *непрозрачный*. В непрозрачном режиме SNMP-менеджер направляет запрос не серверу, а шлюзу SNMP, используя заранее известный порт. Прикладной шлюз производит фильтрацию запроса, в случае успешного результата проверки, отправляет его агенту. Адрес агента определяется самим шлюзом на основе

настроек, выполненных администратором МЭ «Z-2». Прозрачный режим работы шлюза SNMP реализуется только совместно с пакетным фильтром (раздел 5.2.3 на стр. 57). В прозрачном режиме возможна маршрутизация запросов от SNMP-менеджера к нескольким SNMP-агентам.

6.13.3.1 Настройки резидентного процесса

Настройки резидентного процесса осуществляются в разделе `Настройки процесса` (см. рисунок 6.15):

Максимальное количество состояний SNMP — число одновременно обрабатываемых SNMP-запросов (посланных запросов, на которые ожидается ответ);

Имя владельца процесса — имя пользователя — владельца процесса шлюза SNMP;

Название группы процесса — имя группы пользователя — владельца процесса шлюза SNMP;

Предельное время ожидания ответа на запрос (в секундах) — время ожидания ответа на посланный запрос, по истечении которого ответ считается не дошедшим;

Интерфейс, на котором будут приниматься соединения — имя интерфейса, на котором процесс ожидает входящие соединения; можно выбрать из списка имя конкретного интерфейса или пункт «any», в этом случае процесс будет обрабатывать входящие соединения на всех интерфейсах;

Порт для входных данных (DATA) — номер (алиас) порта, на котором шлюз будет ожидать поступления данных, передаваемых SNMP-агентом в ответ на запрос;

Порт для оповещения о событиях (TRAP) — номер (алиас) порта, на котором `snmp-gw` будет ожидать поступления данных, передаваемых SNMP-агентом в ответ на запрос;

Интерфейс для обработки запросов в прозрачном режиме — имя интерфейса, на котором будут обрабатываться входящие запросы в прозрачном режиме; обычно для обеспечения соединения типа «один менеджер — много агентов» в качестве имени следует указать имя интерфейса, соединенного с сетью, в которой находится SNMP-менеджер.

Режим журналирования — режим, в котором будет производиться запись логов в журнал МЭ: `INFO` — информационные сообщения, `DEBUG` — сообщения, содержащие отладочную информацию, `WARNING` — предупреждения.

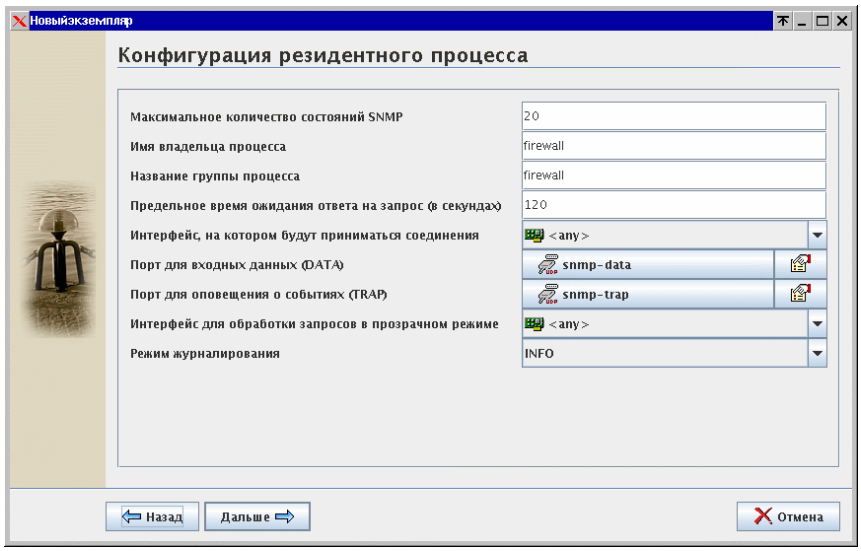

Рис. 6.15: Настройки резидентного процесса для прикладного шлюза SNMP

6.13.3.2 Права доступа и фильтрация

Настройка правил фильтрации для доступа к прикладному шлюзу протокола SNMP производится с помощью раздела графического интерфейса `Политика`. Отдельные правила отображаются графическим интерфейсом пользователя в виде подразделов. Правило доступа привязано к IP-адресу/порту клиента. Более подробно этот механизм описан в разделе 6.2 на стр. 73.

Правило доступа прикладного шлюза SNMP состоит из:

Входящие правила — правила контроля входящих соединений (пакетов);
Исходящие правила — правила контроля исходящих соединений (пакетов);
Опции прокси — опции правила доступа:
- **Хост назначения по умолчанию** — задает хост, на который передаются датаграммы в «непрозрачном» режиме;
- **Порт данных** — задает порт, на который передаются пакеты запросов в «непрозрачном» режиме;
- **Порт Trap** — задает порт, на который передаются пакеты SNMP-TRAP в «непрозрачном» режиме;

Enabled SNMP-communities — определяет список SNMP-сообществ, обращение к которым разрешено. В каждой строке указывается имя только одного сообщества;
Протоколы — с помощью данной вкладки задается один из трех возможных режимов фильтрации SNMP-запросов по версии протокола; возможно разрешить прохождение только одной версии протокола SNMP или обоих поддерживаемых версий; «разрешаемые» версии протокола отмечаются флажком (check-box); установить оба флажка можно с помощью кнопки «Выбрать все»;
Разрешенные команды с помощью данной вкладки определяется список разрешенных команд протокола SNMP. Прикладной шлюз протокола SNMP будет пропускать только те запросы, в которых используются разрешенные команды.

Фильтрация по IP-адресу/порту назначения указывается, как и у остальных прикладных шлюзов, во вкладке `Исходящие правила` правил фильтрации.

6.13.4 Пример конфигурации

Предположим, что для целей мониторинга работы серверов, подключенных в сеть *DMZ*, системный администратор использует протокол SNMP. В целях безопасности посылка SNMP-запросов допустима только с рабочего места администратора (компьютер Z-2_GUI/172.16.10.1) в локальную сеть *DMZ*.

Посылка SNMP-запросов во внешние сети или из внешней сети не допускается.

Предположим, что SNMP-агенты на серверах сети DMZ входят в SNMP-«сообщество» *dmz* и SNMP-протокол применяется только для мониторинга состояния серверов, то есть модификация SNMP-переменных запрещена. Версия SNMP-протокола не ограничивается.

Предположим также, что прием уведомлений (SNMP-traps) от SNMP-агентов сети DMZ запрещается, так как, по соображениям безопасности, из сети DMZ запрещен любой исходящий трафик.

Для решения этой задачи средствами МЭ «Z-2» следует выполнить следующие действия:

1. в правилах пакетного фильтра следует запретить прохождение SNMP-пакетов на внешнем интерфейсе (iprb0);
2. в правилах контроля входящих соединений (пакетов) следует определить правило, разрешающее прохождение запросов с адреса 172.16.10.1 (Z-2_GUI);
3. в правилах контроля исходящих соединений (пакетов) следует определить правило, разрешающее прохождение запросов на адреса сети *DMZ*;
4. в списке наименований SNMP-«сообществ» следует указать *dmz*;

5. в списке допустимых протоколов следует разрешить прохождение SNMP-запросов обоих поддерживаемых протоколов;
6. в списке команд SNMP-протокола следует запретить команды SET, TRAP, TRAP2;
7. активировать на МЭ «Z-2» прикладной шлюз SNMP.

6.14 Прикладной шлюза протокола Oracle SQL*Net

6.14.1 Краткие сведения о протоколе Oracle SQL*Net (Net8)

Протокол Oracle SQL*Net является «транспортным» протоколом, разработанным компанией Oracle для обмена данными между клиентами и серверами СУБД Oracle. В настоящее время используется восьмая версия протокола SQL*Net, получившая название Net8. В дальнейшем названия SQL*Net и Net8 будут использоваться как синонимы.

Протокол SQL*Net может использовать множество сетевых и транспортных протоколов (по классификации OSI), в частности TCP/IP, DECnet, IPX. Прикладной шлюз протокола SQL*Net может осуществлять фильтрацию протокола SQL*Net, только если клиенты и серверы СУБД Oracle используют сеть TCP/IP в качестве опорной.

С точки зрения стека протоколов TCP/IP, поток данных протокола SQL*Net проходит «внутри» одного или двух TCP-соединений. Соединение всегда инициируется клиентом. Клиентом соединения может быть как oracle-клиент (например, утилита sqlplus), так и другой oracle-сервер. Oracle-сервер может открыть соединения с другим oracle-сервером, например, для репликации базы данных.

СУБД Oracle может иметь сложную сетевую структуру и состоять из нескольких серверов и прикладных шлюзов собственного производства (Connection Manager). Клиент, обратившись к одному из серверов (или к прикладному шлюзу Oracle Connection Manager), может быть перенаправлен к другому серверу. Запрос на перенаправление соединения осуществляется посылкой специального пакета (REDIRECT-пакет) со стороны сервера. В redirect-пакете передается адрес и номер TCP-порта сервера, с которым клиент должен установить соединение для работы с базой данных.

Посылка запроса на перенаправление соединения или использование одного соединения зависит от конфигурации клиента и сервера. Информацию о настройках сетевых соединений протокола Net8 и дополнительная информация о протоколе SQL*Net Net8 содержится в фирменной документации Oracle, в частности, в «Oracle Net8 Administrator's Guide» (Release 8.0, фирменный номер документа A58230-01).

6.14.2 Описание прикладного шлюза протокола Oracle SQL*Net

Прикладной шлюз протокола SQL*Net(Net8) является компонентом ПО МЭ «Z-2» и позволяет:

- производить фильтрацию сетевого трафика по адресам и номерам портов источника и приемника соединения;
- ограничивать доступ к серверу СУБД Oracle для пользователей на основании списка «разрешенных» имен пользователей.

Прикладной шлюз Net8 функционирует только в прозрачном режиме совместно с пакетным фильтром. Это ограничение связано с особенностью обработки протокола SQL*Net прикладным шлюзом. Благодаря этой особенности реализации прикладного шлюза Net8, перенастройка клиентов и серверов СУБД Oracle при использовании Net8 не требуется.

Прикладной шлюз Net8 ожидает входящих соединений на двух портах. Один порт (обычно-1521) используется для обработки обычных входящих соединений, а другой — для получения и обработки «перенаправленных» соединений.

Опишем схему функционирования прикладного шлюза Net8 более подробно. Прикладной шлюз Net8 ожидает входящих соединений от клиента на порте для входящих соединений.

Пакеты устанавливаемого соединения с помощью правил перенаправления пакетного фильтра попадают на порт Net8 и обрабатываются прикладным шлюзом. Если адреса и номера портов клиента и сервера подпадают под разрешающие правила политики безопасности, то прикладной шлюз устанавливает соединение с сервером и начинает транслировать пакеты.

Правила перенаправления для пакетного фильтра создаются при запуске прикладного шлюза на основании конфигурационных параметров, а именно: номера порта для входящих соединений и списка серверов Oracle.

Прикладной шлюз Net8 просматривает содержимое пакетов протокола SQL*Net и находит пакет, определяющий начало соединения SQL*Net (TNS-CONNECT). На основании данных, содержащихся в пакете TNS-CONNECT (в частности, имени пользователя), прикладной шлюз либо разрывает соединение с клиентом, либо продолжает транслировать пакеты.

При появлении REDIRECT-пакета от сервера Oracle (TNS-REDIRECT) прикладной шлюз генерирует и загружает в пакетный фильтр правило перенаправления для предполагаемого соединения используя адрес и номер порта сервера из REDIRECT-пакета и адрес клиента, соединение от которого обрабатывается. При этом перенаправление соединения осуществляется на порт для перенаправленных соединений. Данные о предполагаемом «перенаправленом» соединении заносятся во внутреннюю таблицу, общую для всех процессов Net8.

Процесс установления перенаправленного соединения аналогичен процессу установления основного соединения, с той лишь разницей, что при получении перенаправленного соединения, процесс Net8 ищет по внутренней таблице данные о предполагаемом перенаправленом соединении, помещенные в нее на этапе обработки REDIRECT-пакета. Если в таблице нашлась запись о запросе на установление перенаправленного соединения, то прикладной шлюз устанавливает соединение с сервером и начинает трансляцию пакетов. Если данных о перенаправленом соединении в таблице не нашлось, то соединение с клиентом закрывается. После установки перенаправленного соединения дальнейший анализ содержимого пакетов не производится. Обработка основного и перенаправленного соединения осуществляется разными процессами Net8. Более того, после установления перенаправленного соединения, основное соединение между клиентом и сервером не закрывается сразу и прикладной шлюз Net8 обрабатывает два соединения одной сессии.

После завершения перенаправленного соединения прикладной шлюз Net8 удаляет redirect-правило пакетного фильтра.

6.14.3 Настройки резидентного процесса

Прикладной шлюз Net8 реализован по `preforked`-схеме, то есть одновременно запускаются несколько процессов, при этом каждый процесс обслуживает одно соединение. При повышении нагрузки (увеличении числа соединений) автоматически запускаются дополнительные процессы. Подробно preforked-схема работы прикладных шлюзов изложена в главе 6.1.2 на стр. 70.

Кроме общих настроек, группа настроек резидентного процесса содержит следующие настроечные параметры (рисунок 6.16):

Интерфейс для перенаправленных соединений — задает имя интерфейса, на котором прикладной шлюз ожидает входящие перенаправленные соединения. Значением параметра следует установить имя интерфейса, подключенного к сегменту сети в которой находятся Oracle-клиенты.

Порт для перенаправленных соединений — указывает номер порта, на котором прикладной шлюз ожидает перенаправленные соединения.

6.14.4 Список пользователей Oracle

В списке пользователей Oracle (раздел графического интерфейса `Пользователи шлюза`) указываются имена пользователей, от имени которых запускаются клиенские программы (login names).

Сетевая архитектура Oracle использует два множества имен:

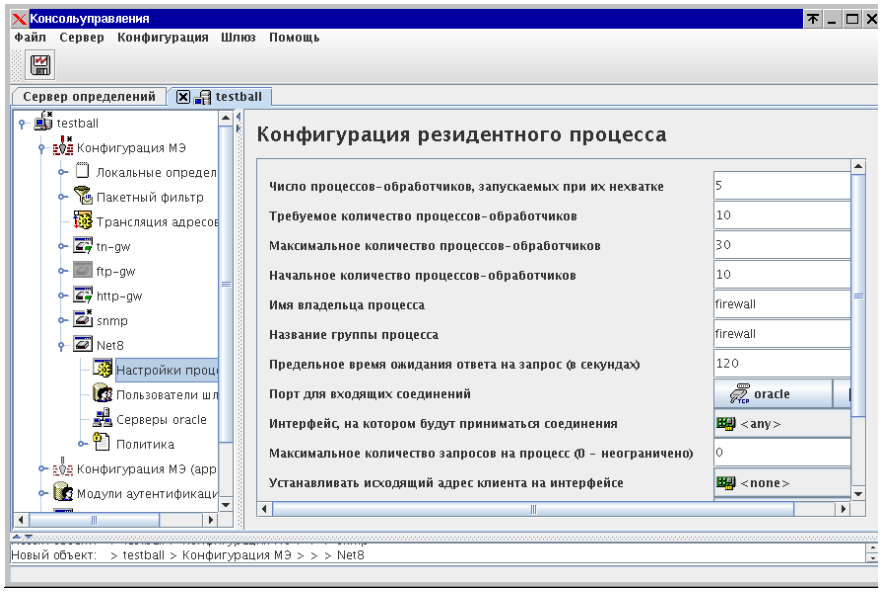

Рис. 6.16: Настройки резидентного процесса

1. имена регистрации пользователей клиентских программ, используемые на сетевом уровне, например для установления защищенного соединения;
2. имена пользователей базы данных, используемые для разграничения доступа.

Прикладной шлюз Net8 контролирует только имена регистрации пользователей.

6.14.5 Список серверов Oracle

В списке серверов (раздел графического интерфейса `Серверы oracle`) должны быть перечислены все серверы Oracle, с которыми может быть установлено соединение. В списке следует указать имя (или адрес) сервера и номер порта, на котором ПО СУБД Oracle ожидает входящие соединения.

При настройке прикладного шлюза Net8 следует указать адреса всех используемых серверов, поскольку в начале работы прикладной шлюз Net8 использует адреса из списка серверов для генерации служебных правил для пакетного фильтра.

6.15 Утилита fwctl

Программа fwctl предназначена для управления процессами запуска/остановки МЭ Z-2 из командной строки. Она расположена в директории */opt/JETmserv/sbin*. Программа используется сервером управления mserv и стартовыми скриптами. Утилита `fwctl` позволяет выполнять:

- запуск прикладного шлюза;
- останов прикладного шлюза;
- перезагрузку текущей конфигурации прикладных шлюзов и пакетного фильтра;
- запуск всех активных прикладных шлюзов и загрузку текущих правил фильтрации и трансляции адресов в пакетный фильтр;
- останов всех активных прикладных шлюзов и сброс правил фильтрации и трансляции адресов в пакетном фильтре;
- смену текущего конфигурационного набора, перегенерацию правил фильтра и шлюза.

6.15.1 Синтаксис вызова программы fwctl

Общий синтаксис вызова программы `fwctl` выглядит следующим образом:

```
/opt/JETmserv/sbin/fwctl [start <proxy_binary> <proxy_instance>|stop <proxy_in
```

6.15.2 Список опций программы fwctl

startall
Запустить все сконфигурированные прикладные шлюзы и загрузить правила пакетного фильтра. Скрипт включает режим маршрутизации ОС Solaris.

stopall
Остановить все активные прикладные шлюзы, сбросить текущие правила фильтрации, правила трансляции адресов в пакетном фильтре и выключить режим марштутизаци ОС Solaris.

start <шлюз> <имя>
Запустить экземпляр прикладного шлюза, указанный в параметре <имя>. Параметр <шлюз> указывает имя исполняемого файла этого шлюза, например `http_gw` для прикладного шлюза протокола HTTP.

stop <имя>
Остановить экземпляр прикладного шлюза, указанный в параметре <proxy_instance>.

reload
Перезагрузить все прикладные шлюзы и правила фильтрации и трансляции адресов в пакетном фильтре.

create <имя>
Создать новый конфигурационный набор.

change <имя>
Переключить активный конфигурационный набор. Новую конфигурацию необходимо активизировать командой fwctl reload.

cfgen перегенерирует правила пакетного фильтра и прикладных шлюзов.

confgen перегенерирует правила пакетного фильтра и прикладных шлюзов.

Глава 7

Сервер аутентификации

Сервер аутентификации и авторизации обеспечивает проверку аутентичности, а также прав доступа пользователей при их обращении к прикладным шлюзам. Собственно аутентификация и авторизация осуществляется встраиваемыми аутентификационными модулями (PAM модулями). Их количество может быть произвольным.

В настоящее время с МЭ «Z-2» поставляются PAM-модули для следующих схем аутентификации:

Plain password — модуль`pam_db`
S/Key — модуль`pam_skey`
Radius — модуль`pam_radius_auth`
Windows NT domain — модуль`pam_smb`

Использование технологии PAM-модулей позволяет:

- динамически добавлять новые схемы аутентификации без изменения кода сервера аутентификации;
- использовать для различные технологии аутентификации для различных пользователей;
- осуществлять общее управление учетными данными пользователей.

Процесс аутентификации на сервере аутентификации МЭ «Z-2» происходит следующим образом:

- Прикладной шлюз обращается к серверу аутентификации и сообщает имя пользователя (`login`), которого необходимо аутентифицировать.
- Сервер аутентификации производит поиск данного пользователя в своей базе данных пользователей. Если такой пользователь найден, производится проверка того, что пользователь не заблокирован и ему разрешен доступ. Если проверка прошла успешно, управление передается на соответствующий ему PAM-модуль, который и производит аутентификацию этого пользователя. Результат аутентификации сервер аутентификации передает прикладному шлюзу;
- Если пользователь не найден, управление передается PAM-модулю, используемому по умолчанию, который определен в настройках сервера аутентификации.

Таким образом, в базе данных пользователей сервера аутентификации не содержится непосредственно аутентификационной информации пользователей (пароли, токены и т.п.). Вся подобная информация содержится в базе пользователей PAM-модулей и в общем случае может находится вне МЭ.

7.1 Краткие сведения об архитектуре PAM

PAM это используемая в ОС Solaris (и некоторых других UNIX-системах) схема подключения аутентификационных модулей. *PAM-модуль* представляет собой динамически погружаемый объектный файл, реализующий определенных механизм аутентификации.

PAM-модуль может реализовать следующие функции:

Функция	**Описание**
account	проверка учетной записи пользователя на существование и доступность в данный момент
auth	аутентификация пользователя
session	управление сеансом работы пользователя
password	смена аутентификационной информации (обычно пароля)

Конфигурация PAM-модулей задается в системном конфигурационном файле `/etc/pam.conf`, содержащем список сервисов, в соответствии которым ставится PAM-модуль, например:

```
login auth required /usr/lib/security/$ISA/pam_unix.so.1
```

Параметр	**Описание**
login	имя имя сервиса, использующего PAM-модуль
auth	используемая функция модуля
required	действие по результатам проверки: • `requisite` — немедленный отказ в аутентификации в случае отрицательного ответа модуля; • `required` — отложенный отказ в аутентификации в случае отрицательного ответа модуля. Т.е. в любом случае осуществляется переход на следующий модуль в цепочке, но по окончанию проверок выдается отказ в аутентификации; • `sufficient` — в случае успешной аутентификации немедленно дается положительный ответ. Действие имеет меньший приоритет, чем `required`; • `optional` — необязательный модуль. Ответ модуля имеет значение только при отсутствии других модулей в цепочке. В остальных случаях ответ игнорируется.
/usr/lib/security/$ISA/pam_unix.so.1	полный путь к модулю и его аргументы. Набор аргументов модуля в общем случае определяется самим модулем. Обычно PAM-модули поддерживают опцию `use_authtok`, предлагающую модулю использовать последний переданный предыдущему модулю в цепочке пользовательский ввод (используется для избежания множественного ввода пароля по ходу цепочки), и опция `debug`, включающая режим отладки модуля.

7.2 Настройки сервера аутентификации

Настройки сервера аутентификации состоят из:

- настроек резидентного процесса (раздел графического интерфейса `Настройки процесса`)
- настроек базы данных пользователей (раздел графического интерфейса `Настройки базы данных пользователей`)

7.2.1 Настройки резидентного процесса

Сервер аутентификации работает по `preforked`-схеме, т.е. одновременно в системе запущено множество процессов сервера аутентификации, слушающих и обслуживающих запросы на аутентификацию. Одновременно один процесс обслуживает одного клиента (раздел *«Preforked схема»*). Настройка резидентного процесса показана на рисунке 7.1.

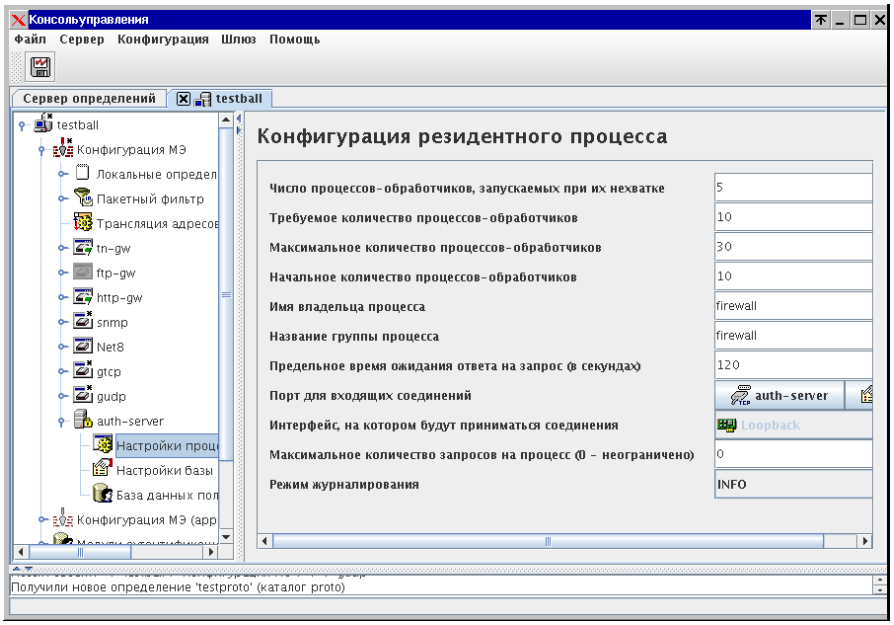

Рис. 7.1: `Конфигурация резидентного процесса`

7.2.2 Настройка базы данных пользователей

В таблице 7.3 приведены параметры базы данных пользователей.

Таблица 7.3: **Параметры базы данных пользователей**

Параметр	Описание
Имя файла с базой данных пользователей	Имя файла базы данных пользователей. Порядок работы с базой описан в разделе 7.3 на стр. 133.

Продолжение следует

. . . продолжение таблицы 7.3.

Имя цепочки модулей аутентификации по умолчанию	Имя сервиса для аутентификации, использующегося по умолчанию. В случае отсутствия данного атрибута пользователи, для которых имя сервиса не определено, или пользователи, не заведенные в базе пользователей, получают отказ в аутентификации.
Блокировка пользователей по умолчанию	Состояние блокировки пользователя по умолчанию. Значение `On` (отмеченный переключатель) означает, что по умолчанию только что созданный пользователь заблокирован.
Часы доступа по умолчанию	Время допуска пользователей по умолчанию. Формат этого поля совпадает с форматом поля в базе пользователей и описан в таблице 7.4. Для задания времени вызывается специальный редактор, где отмечаются нужные часы.
По умолчанию запрещать менять пароль	Возможность смены пользователем собственного пароля.

Настройка параметров базы данных пользователей показана на рисунке 7.2.

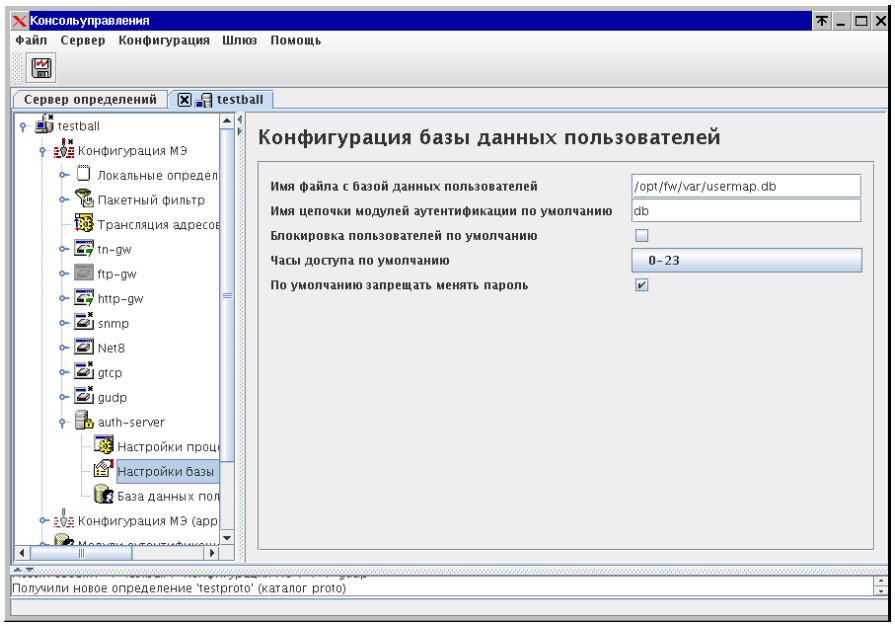

Рис. 7.2: Конфигурация базы данных пользователей

7.3 База данных пользователей

База данных пользователей содержит информацию, необходимую для аутентификации и работы пользователей (рисунок 7.3). Одна строка в таблице базы соответствует одному пользователю. Ключом является имя пользователя, а полями:

```
login autoblock pam blocked pwdchange hours
```

Описание полей приведено в таблице. 7.4. Значения полей по умолчанию определяются при настройке сервера аутентификации. Более подробно этот вопрос рассмотрен в разделе 7.2.2 на стр. 131.

Таблица 7.4: **Поля записи в базе данных пользователей**

Поле	**Описание**	**Значение по умолчанию**
login	Имя (логин) пользователя	Не имеет
autoblock	Состояние пользователя после исчерпания лимита неудачных попыток аутентификации. + означает то, что пользователь был заблокирован после исчерпания лимита неудачных попыток аутентификации. В дальнейшем разрешить пользователю вход может только администратор межсетевого экрана.	
pam	Имя сервиса аутентификации. Сервис db обеспечивает аутентификацию по логину/паролю через PAM pam_db, сервис skey обеспечивает аутентификацию одноразовыми паролями через PAM pam_skey.	Определяется параметром Имя цепочки модулей аутентификации по умолчанию.
blocked	Состояние блокировки. + пользователь заблокирован, – пользователь разблокирован. Пробел — значение по умолчанию.	Определяется параметром Блокировка пользователей по умолчанию.
pwdchange	Блокировка возможности самостоятельной смены пароля пользователем. + смена пароля пользователем заблокирована, – смена пароля разрешена. Пробел — значение по умолчанию.	Определяется атрибутом По умолчанию запрещать менять пароль.
hours	Время допуска пользователя. Для задания времени допуска используется специальный редактор.	Определяется параметром Часы доступа по умолчанию.

Когда пользователь израсходует лимит неудачных попыток аутентификации, его доступ к межсетевому экрану будет заблокирован. В графическом интерфейсе в этом случае в графе autoblock для данного пользователя будет установлено значение +. Графический интерфейс получает сведения о состоянии пользователей только в момент инициализации, поэтому пользователи, заблокированные во время сессии администрирования, отображены не будут.

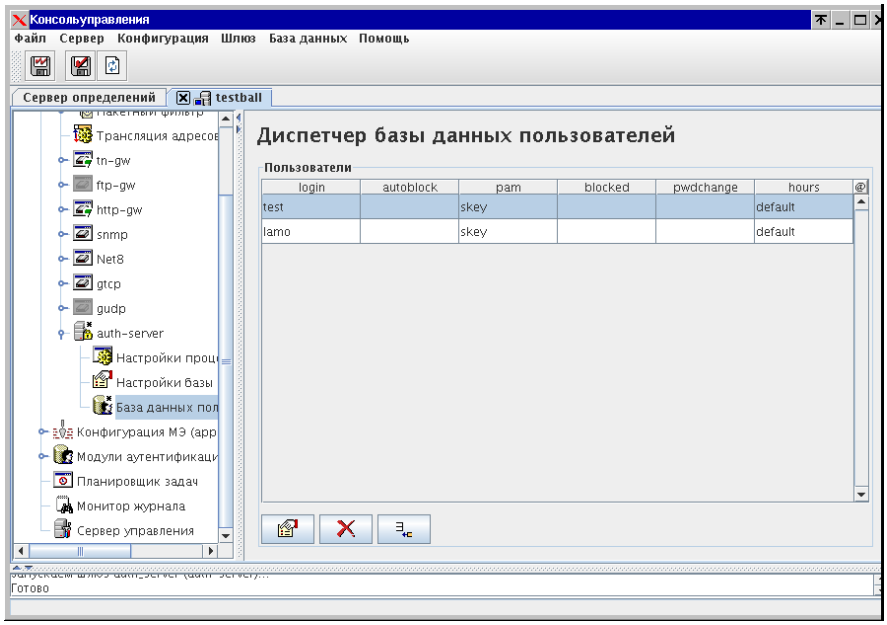

Рис. 7.3: Настройки базы данных пользователей

7.4 Модуль авторизации pam_radius_auth

Модуль аутентификации pam_radius представляет собой разделяемую библиотеку libpam_radius_auth.so и предназначается для выполнения аутентификации пользователей МЭ «Z-2» посредством внешнего сервера авторизации и учета ресурсов, реализующего протокол RADIUS. Модуль может загружаться как сервером аутентификации МЭ «Z-2», так и штатными программами аутентификации OS Solaris(например login или telnetd).

Разделяемая библиотека libpam_radius_auth.so копируется в директорию /usr/lib/security в процессе установки ПО МЭ «Z-2».

Настройка модуля аутентификации pam_radius заключается в модификации файла описания существующих в системе pam-модулей /etc/pam.conf и создании файла с описанием параметров соединения с RADIUS-сервером.

Описание настройки RADIUS-сервера следует искать в составе документации, поставляемой с используемым ПО RADIUS-сервера.

7.4.1 Файл /etc/pam.conf

В конец файла /etc/pam.conf следует дописать строку:

```
radius auth required /usr/lib/security/$ISA/libpam_radius_auth.so <опции>
```

Необязательное поле <опции> содержит список параметров загрузки разделяемой библиотеки, которые будут описаны ниже.

7.4.2 Описание параметров загрузки разделяемой библиотеки

Разделяемая библиотека libpam_radius_auth.so распознает следующие параметры загрузки (указываются в файле /etc/pam.conf):

debug — выводить отладочные сообщения в файл /var/adm/messages через системную службу syslog,

use_first_pass — использовать пароль, переданный предыдущему в списке аутентификации pam-модулю. Если «предыдущего» пароля не существует (передан пустой или модуль pam_radius — первый в списке), то завершить аутентификацию со статусом «не успешно». Если пароль получен, то передать его RADIUS-серверу и передать pam-менеджеру статус завершения аутентификации от сервера.

try_first_pass — использовать пароль, переданный предыдущему в списке аутентификации pam-модулю. Если «предыдущего» пароля не существует(передан пустой или модуль pam_radius — первый в списке), то перезапросить пароль у пользователя. Полученный (от предыдущего модуля или от пользователя) пароль передать RADIUS-серверу. Этот алгоритм работы модуля используется по-умолчанию.

client_id=bar — этот параметр задает «имя сервера доступа» (NAS-Identifier). В терминах протокола RADIUS, NAS-Identifier соответствует имени клиента. Если параметр не задан, то в качестве имени клиента передается имя сервиса, вызвавшего модуль. Если параметр указан в виде 'client_id=', то pam-модуль передает RADIUS-серверу пустое имя клиента.

skip_passwd — Передать RADIUS-серверу пароль, переданный предыдущему в списке аутентификации pam-модулю. Если «предыдущий» модуль не передал пароль (передан пустой или модуль pam_radius — первый в списке), то не запрашивать пароль у пользователя и передать RADIUS-серверу «пустой» пароль. Если от сервера придет запрос на введение пароля (Access-Challenge),

то выдать приглашение пользователю и передать ответ пользователя RADIUS-серверу.

conf=foo — использовать указанный файл описания параметров соединения (конфигурационный файл). По-умолчанию модуль использует файл /etc/raddb/server.

client_id=bar — этот параметр задает «имя сервера доступа» (NAS-Identifier). В терминах протокола RADIUS, NAS-Identifier соответствует имени клиента. Если параметр не задан, то в качестве имени клиента передается имя сервиса, вызвавшего модуль. Если параметр указан в виде 'client_id=', то рам-модуль передает RADIUS-серверу пустое имя клиента.

use_authtok — этот параметр форсирует передачу пароля, стороннему модулю. Опция необходима при использовании модулей проверки «надежности» пароля (например, пароль сначала передается в модуль cracklib, а, затем — RADIUS-серверу).

accounting_bug — этот параметр задает специальный режим совместимости со старыми версиями RADIUS-серверов (например, Livingston 1.16). Параметр отключает верификацию контрольных сумм некоторых ответов от сервера. В настоящее время, старые версии серверов практически не используются.

7.4.3 Файл параметров соединения с RADIUS–сервером

Для указания адреса удаленного RADIUS-сервера следует создать файл описания параметров соединения. По умолчанию, файл имеет имя server и размещается в каталоге */etc/raddb*. При использовании модуля `pam_radius` совместно с сервером аутентификации МЭ «Z-2» следует установить для файла */etc/raddb/server* имя владельца firewall и имя группы-владельца firewall. Так же, для файла */etc/raddb/server* следует установить маску доступа 0600.

Файл */etc/raddb/server* имеет текстовый формат и содержит записи следующего вида:

```
server[:port] shared_secret [timeout],
```

где:

Поле <server> содержит доменное имя (адрес) RADIUS-сервера

Поле <port> содержит номер порта, на котором удаленный RADIUS-сервер ожидает приходящие запросы. По-умолчанию используется порт 1812.

Поле <shared_secret> содержит пароль, используемый для кодирования запросов к RADIUS-серверу и его ответов. Пароль представляет собой текстовую строку без пробелов и записывается в открытом виде.

Поле <timeout> определяет время (в секундах) ожидания ответа от RADIUS-сервера. По-умолчанию, время ожидания — 3с.

Если файл */etc/raddb/server* содержит несколько описаний RADIUS-серверов, то модуль `pam_radius` будет последовательно обращаться к каждому указанному серверу. Ответ первого «ответившего» сервера будет считаться результатом аутентификации и будет передан вызвавшему модуль приложению. Ответ от сервера ожидается течении указанного промежутка времени (поле <timeout>).

7.5 Модуль аутентификации pam_smb

Модуль аутентификации pam_smb представляет собой разделяемую библиотеку *libpam_smb.so* и предназначается для выполнения аутентификации пользователей МЭ «Z-2» с использованием протокола авторизации, используемого в сетях Windows (NT domain authentification).

Модуль может загружаться как сервером аутентификации МЭ «Z-2», так и штатными программами аутентификации OS Solaris (например, login или telnetd).

Настройка модуля аутентификации pam_smb заключается в модификации файла описания существующих в системе pam-модулей */etc/pam.conf* и создании файла */etc/pam_smb.conf*, содержащий параметры соединения с контроллерами доменов сети Windows.

7.5.1 Конфигурационный файл */etc/pam_smb.conf*

Файл */etc/pam_smb.conf* имеет текстовый формат и должен содержать следующие три строки:

- Имя домена, в котором требуется авторизовать пользователя;
- Имя главного контроллера домена (Primary Domain Controller — PDC);
- Имя резервного контроллера домена (Backup Domain Controller — BDC).

Модуль pam_smb не требует, чтобы на серверах, указанных в файле */etc/pam_smb.conf* исполнялась именно ОС WindowsNT (Windows 2000). Достаточно, чтобы ПО, исполняемое на указанных серверах было способно обрабатывать запросы на авторизацию протоколом SMB.

7.5.2 Описание изменений файла */etc/pam.conf*

В конец файла */etc/pam.conf* следует дописать строку:

```
other auth required /usr/lib/security/$ISA/libpam_smb.so <опции>
```

Необязательное поле <опции> содержит список параметров загрузки разделяемой библиотеки, которые будут описаны ниже.

7.5.3 Описание параметров загрузки (командной строки) разделяемой библиотеки

Разделяемая библиотека libpam_smb.so распознает следующие параметры загрузки (указываются в файле */etc/pam.conf*):

debug — выводить отладочные сообщения в файл /var/adm/messages через системную службу syslog,

use_first_pass — использовать пароль, переданный предыдущему в списке аутентификации pam-модулю. Если «предыдущего» пароля не существует (передан пустой пароль или модуль pam_smb — первый в списке), то завершить аутентификацию со статусом «не успешно». Если пароль получен, то передать его PDC(BDC) и передать pam-менеджеру статус завершения аутентификации от сервера.

nolocal — не использовать при верификации пары имя пользователя/пароль локальную базу авторизации (файл */etc/passwd*).

Глава 8

Система обнаружения атак SNORT IDS

МЭ «Z-2» версий 1.8 и выше поддерживает интеграцию с системой обнаружения атак Snort IDS.

Графический интерфейс Z-2 предоставляет базовые функции управления Snort (смотри книгу *«Универсальный графический интерфейс 2.4 Руководство администратора»*). Полноценное администрирование `Snort`, поддержка базы сигнатур атак и т.п. осуществляется системным администратором и не является функцией Z-2.

Интеграция МЭ «Z-2» со `SNORT IDS` включает в себя следующие возможности:

- Создание базовой конфигурации IDS через GUI Z-2, перезапуск Snort
- Просмотр предупреждающих сообщений (Alert) в `Log Viewer` (GUI Z-2)
- Возможность реагирования на атаку:
  1. `log` - запись Alert в журнал Z-2;
  2. `block` - блокировка источника атаки;
  3. `ignore` - игнорирование атаки;
- Возможность задания «дружественной» сети, атаки из которой будут игнорироваться.

Глава 9

Протоколирование информации

МЭ «Z-2» использует подсистему протоколирования `syslogd` для записи своих сообщений. По умолчанию предполагается, что `syslogd` производит запись в файл */var/adm/messages*. Просмотр файла осуществляется с помощью штатных утилит операционной системы. При помощи специальной подсистемы GUI (раздел графического интерфейса `Монитор журнала`) можно просматривать записи системного журнала в режиме реального времени.

9.1 Формат записи в журнале протоколируемых событий прикладных шлюзов

МЭ «Z-2» поддерживает три типа записей:

1. Информационные сообщения.
2. Предупреждения.
3. Сообщения об ошибках.

Информационные сообщения и предупреждения имеют унифицированную структуру:

```
<Protocol> <subject> <src-addr>[:<src-port>] -> [<dst-addr>[:<dst-port>]] <action> [.<Rea
```

Поля имеют следующее значения:

Protocol — один из следующих прикладных протоколов:
1. FTP
2. HTTP
3. SMTP
4. TELNET
5. TCP
6. UDP
7. SNMP

subject — краткая характеристика произошедшего события.
src-addr — IP-адрес (доменное имя) клиента прикладного шлюза.
dst-addr — ip-адрес(доменное имя) удаленного ресурса, с которым прикладной шлюз устанавливает соединение.
src-port — номера портов клиента ресурса с которого устанавливается соединение.
dst-port — номера портов удаленного ресурса на который устанавливается соединение.
action — действие, результатом которого явилось протоколируемое событие.
.Reason — Дополнительное поле. Возможная причина произошедшего события.
.AddtlInfo — Дополнительное поле. Комментарий к произошедшему событию.

Сообщения об ошибках не стандартизируются. Все они протоколируются при завершении работы программы.

9.1.1 Информационные сообщения

Каждый прикладной шлюз выдает информационное сообщение в следующих случаях:

1. Подсоединение шлюза к удаленному ресурсу по запросу клиента.
2. Рассоединение шлюза с удаленным ресурсом/клиентом.
3. Передача команды протокола, протоколирование которой затребовано администратором.

Во всех сообщениях, приведенных ниже действуют следующие обозначения:

```
<Protocol> ::= "FTP"    |
       "HTTP"   |
       "SNMP"   |
       "SMTP"   |
       "TELNET" |
       "TCP"    |
       "UDP"
```

```
<src-addr> - ip-адрес(доменное имя) клиента прикладного шлюза.
<dst-addr> - ip-адрес(доменное имя) удаленного ресурса, с которым
             прикладной шлюз устанавливает соединение.
<src-port>/<dst-port> - номера портов клиента/удаленного ресурса
                        с/на которые устанавливается соединение.
```

9.1.1.1 Подсоединение шлюза к удаленному ресурсу по запросу клиента

Для шлюзов протокола TCP (прикладные шлюзы протоколов FTP, HTTP, SMTP, telnet, универсальный прикладной шлюз TCP):

```
<Protocol> connection <src-addr>:<src-port> ->
<dst-addr>:<dst-port> established.
```

Для универсального прикладного шлюза UDP:

```
UDP transmition from <src-addr>:<src-port> -> <dst-addr>:<dst-port> enabled.
```

Для прикладного шлюза протокола SNMP:

```
SNMP transmition from <src-addr>:<src-port> -> <dst-addr>:<dst-port> enabled.
```

9.1.1.2 Рассоединение шлюза с удаленным ресурсом/клиентом

Для шлюзов протокола TCP (прикладные шлюзы протоколов FTP, HTTP, SMTP, telnet, универсальный прикладной шлюз TCP):

```
<Protocol> connection <src-addr>:<port> -> <dst-addr>:<port> closed.
In: <bytes>. Out: <bytes>
```

Для универсального прикладного шлюза протокола UDP:

```
UDP transmition <src-addr>:<src-port> -> <dst-addr>:<dst-port>
terminated. In: <bytes>. Out: <bytes>
```

Для прикладного шлюза протокола SNMP выдача этого сообщения не предусматривается.

9.1.1.3 Передача команды протокола, протоколирование которой затребовано администратором

```
<Protocol> <Command> command sent<src-addr>:<port> -> <dst-addr>:<port>
```

9.2 Протокол работы пакетного фильтра

Формат записи в файл протокола, производимый модулем `ipmon`, отличается от рассмотренного выше формата записи прикладных шлюзов:

```
Aug  9 23:37:19 snowball ipmon[119]: [ID 702911 local0.warning]
23:37:19.035529 hme0 @0:2 b 192.168.10.122,43206 ->
172.16.10.133,27410 PR tcp len 20 44 -S IN>
```

Запись представляет собой текстовую строку, в которой поля разделены пробелами. Строка начинается записью, выполненной `syslogd`:

```
Aug  9 23:37:19 snowball ipmon[119]: [ID 702911 local0.warning]
```

После которой следует запись `ipmon`:

```
23:37:19.035529 hme0 @0:2 b 192.168.10.122,43206 ->
172.16.10.133,27410 PR tcp len 20 44 -S IN
```

Поля, общие для всех записей имеют следующие значения:

1. Дата регистрации пакета.
2. Время регистрации пакета с точностью до долей секунды.
3. Имя интерфейса, через который производилось считывание/запись пакета. Например, `hme0`.
4. Группа и номер правила, под действие которой подпал данный пакет. Например, `@0:2`.
5. Результат обработки пакета: `p` — пакет пропущен, `b` — пакет заблокирован.
6. Данные об адресе:порте отправителя пакета `->` данные об адресе:порте адресата пакета.
7. Ключевое слово `PR` за которым следует имя протокола. Например: `PR tcp`.
8. Ключевое слово `len` за которым следует длина заголовка пакета и длина всего пакета.

Далее могут следовать одно или несколько полей, специфичных для каждого протокола.

9.3 Формат записи протоколов mserv

Управляющий сервер — mserv имеет формат записи протоколов, несколько отличающийся от рассмотренных ранее. Ниже приведен пример записи:

```
Oct 18 11:22:13 Zhost mserv:proxy: Administrator CN=test client 2,
OU=LPR, O=Jet Infosystems, ST=Central Russia, C=ru [127.0.0.1:8766]
closed connection successfully
Oct 18 11:22:17 Zhost mserv:auth: Administrator CN=test client 2, OU=LPR,
O=Jet Infosystems, ST=Central Russia, C=ru [127.0.0.1:8766] connected
successfully
Oct 18 11:22:17 Zhost mserv:auth: Administrator CN=test client 2, OU=LPR,
O=Jet Infosystems, ST=Central Russia, C=ru [192.168.10.102:5081] got known
PAM modules list
Oct 18 11:22:17 Zhost mserv:auth: Administrator CN=test client 2, OU=LPR,
O=Jet Infosystems, ST=Central Russia, C=ru [192.168.10.102:5081] closed
connection successfully
```

В каждую запись входят следующие поля, разделенные пробелами:

1. Дата и время записи.
2. Имя компьютера, на котором функционирует mserv.
3. Постоянное поле mserv, а затем, через : имя модуля, которой был запрошен.
4. Логин (определяющее имя) пользователя, от которого поступил запрос.
5. Идентификатор пользователя в форме записи LDAP.
6. IP-адрес и порт, откуда поступил запрос к серверу.
7. Результат произведенного действия.

Глава 10

Анализ регистрационной информации

10.1 Использование программы plog

В качестве средства анализа регистрационной информации в МЭ «Z-2» используется программа plog, позволяющая генерировать отчеты на основе протоколируемой информации, получаемой от монитора пакетного фильтра ipmon

Программа plog генерирует два вида отчетов:

Отчет по адресам - отправителям IP-пакетов В данном отчете все пакеты, исходящие с какого либо адреса, просуммированы и представлены в виде таблицы вида:

```
iface action packet-count proto src-port dest-host.dest-port [(flags)]
```

где:

iface имя интерфейса, на который поступил данный пакет
packet-count число поступивших пакетов
proto имя протокола
src-port номер порта, с которого был отправлен данный пакет
dest-host.dest-port адрес машины и номер порта, на который был отправлен данный пакет
flags флаги, установленные в данном пакете. Данное поле по умолчанию не отображается и заполняется только при использовании опции -F

Отчет по адресам - получателям IP-пакетов В данном отчете все пакеты, направленные на какой-либо адрес, просуммированы и представлены в виде таблицы вида:

```
iface action packet-count proto src-port dest-host.dest-port [(flags)]
```

значения полей в этой таблице аналогичны вышеописанным.

Общий синтаксис вызова программы plog выглядит следующим образом:

```
/opt/fw/bin/plog [-nSDF] [-s <имя файла>] [-A  <правило1>, ... ] [<IP адрес>..
```

где:

-n не использовать разрешение IP-адресов в именах хостов
-S Сформировать отчет по адресам — отправителям IP-пакетов
-D Сформировать отчет по адресам — получателям IP-пакетов
-F Показывать в отчете все флаги, установленные в IP-пакете
-s <имя файла> Использовать альтернативную таблицу соответствий между номером и названием порта. Предоставленный файл должен иметь формат, аналогичный файлу /etc/services. Если для номера порта нет соответствия в указанном файле, то будет использован файл /etc/services.
-A <правило1>, ... При генерации отчета использовать только указанные правила. Возможными правилами могут быть:

- pass
- block
- log
- nomatch

<IP-адрес> Если в качестве аргументов указан один или несколько адресов, то отчет будет сформирован только для них. <IP-адрес> может быть задан как символьное имя хоста, IP-адрес или сетевая маска вида `network_address/network_length` (например, `192.168.0.0/24`). Если указанное символьное имя хоста имеет несколько IP-адресов, в все они будут использованы в отчете.

Если в списке опций не присутствует `-D` или `-S`, формируется отчет как по адресам — отправителям, так и по адресам — получателям

10.2 Примеры использования программы plog

```
/opt/fw/bin/plog -S  www.jet.msk.su < /var/adm/messages
```

Получить отчет по всем пакетам, отправленным с хоста www.jet.msk.su

```
/opt/fw/bin/plog -AF block < /var/adm/messages
```

Получить отчеты по адресам — отправителям и адресам — получателям для всех заблокированных пакетов. В отчет включить информацию по флагам, установленным в этих пакетах.

10.3 Программа анализа протоколов прикладных шлюзов logstat.pl

10.3.1 Краткое описание программы logstat.pl

Программа анализа протоколов прикладных шлюзов logstat.pl выполняет синтаксический анализ протоколов, создаваемых прикладными шлюзами, выполняет выборку определенных записей и печатает статистику. В качестве входных файлов утилита logstat.pl предполагает файлы /var/adm/messages.*. Утилита может обработать неограниченное число файлов. Список файлов задается в командной строке и предполагается упорядоченным по времени создания.

Статистика, собираемая программой logstat.pl, включает в себя:

- интервал времени, за который собрана статистика;
- объем принятой/переданной информации;
- число запросов (установленных соединений), обслуженных всеми прикладными шлюзами;
- число запросов (установленных соединений), запрещенных всеми прикладными шлюзами;
- число обработанных записей по категориям (ошибки, предупреждения, информационные сообщения).

Кроме общей статистики, для каждого прикладного шлюза программа выдает списки запрещенных политикой безопасности адресов серверов и клиентов, на которые (с которых) производились попытки установить соединения, списки «наиболее популярных» ресурсов (ip-адреса, URL, почтовые адреса) на которые устанавливались соединения, и другую информацию.

Утилита logstat целиком реализована на языке perl5. Утилита не требует для своей работы установки дополнительных библиотек поддержки (модулей) языка perl.

Для определения списка «активных» в текущей конфигурации МЭ «Z-2» прикладных шлюзов и списка «сущностей» каждого прикладного шлюза утилита logstat.pl использует конфигурационный файл. По умолчанию конфигурационный файл имеет имя logstat.conf и находится в каталоге /opt/fw/etc/. Конфигурационный файл создается управляющим сервером на основании данных, переданных от графического интерфейса администратора.

10.3.2 Запуск программы logstat.pl

Формат командной строки утилиты logstat приведен ниже:

```
logstat.pl [-f <cfg>] [-l <length>] [-s <order>] [-r] <lfile>...
```

где:

<cfg> — имя конфигурационного файла для утилиты (описание см. ниже). По умолчанию '/opt/fw/etc/logstat.conf';
<length> — длина списков адресов/url/файлов в статистике;
<order> — критерий сортировки списков. Принимает следующие значения:
'in' — сортировать по количеству принятых байт;
'out' — сортировать по количеству переданных байт;
'count' — сортировать по числу запросов;
-r — разрешать ip-адреса до имен с помощью DNS. По умолчанию адреса не разрешаются;
<lfile>... — список файлов для анализа.

10.3.3 Формат конфигурационного файла

Конфигурационный файл имеет текстовый формат и состоит из следующих строк:

<Строка формата>
<Строка формата>
<Строка формата> ::= <комментарий> | <описатель> | <пустая строка>

<комментарий> ::= '#'Произвольный текст
<пустая строка> ::= 'LF'
<описатель> ::= <имя шлюза>':'['TAB'|'SPACE']+ <список сущностей>
<имя шлюза> ::= 'http_gw' | 'ftp_gw' | 'smtp_gw' | 'snmp_gw' | 'tn_gw' | 'gudp_gw' | 'gtcp_gw'
<список сущностей> ::= 'имя' | 'имя' <список сущностей>

В одинарные кавычки помещены терминальные символы.

Пример конфигурационного файла приведен ниже:

```
# Это--комментарий
http_gw: http-gw http-gw1 http-gw2
ftp_gw:  ftp-gw ftp-gw1 ftp-gw2
smtp_gw: smtp-gw
snmp_gw: snmp-gw snmp-gw1
tn_gw:   tn-gw
gudp_gw: gudp-gw1 gudp-gw2 gudp-gw3
gtcp_gw: gtcp-gw
```

10.4 Утилита tailstat

Утилита tailstat позволяет просматривать постоянно пополняющиеся файлы (например, файл сообщений */var/adm/messages*) начиная с «конца последнего просмотра».

Алгоритм утилиты tailstat сводится к передаче содержимого файла, заданного по имени в командной строке в стандартный поток вывода. При обнаружении конца файла, утилита tailstat записывает текущую длину файла (величину смещения файлового указателя от начала файла) в файл состояния. При следующем запуске утилиты tailstat с теми же параметрами командной строки, утилита читает созданный ранее файл состояния и производит выдачу содержимого переданного по имени файла начиная с сохраненной позиции. После окончания вывода файл состояния обновляется.

10.4.1 Запуск утилиты tailstat

Запуск утилиты tailstat производится командой:

```
tailstat [-s <state_file>] <file>
```

Аргументами командной строки для утилиты являются:

<file> — обязательный параметр, задающий имя файла;
<state_file> — необязательный параметр, задающий имя «файла состояния».

По умолчанию утилита tailstat создает файл состояния в каталоге */var/tmp*. Файл состояния имеет имя, составленное из имени обрабатываемого файла, его числового идентификатора на файловой системе (i-node) и идентификатора файловой системы. Тем самым обеспечивается уникальность имени файла состояния для всех обрабатываемых файлов, расположенных на локальных файловых системах. На имя файла состояния, переданного пользователем в параметре «-s» командной строки, утилита tailstat не накладывает никаких ограничений.

Глава 11

Подсистема контроля целостности

В качестве средства обеспечения контроля целостности программной и информационной части МЭ «Z-2» используется программный продукт `Tripwire`, позволяющий отслеживать все изменения в наборе файлов и каталогов, определенном системным администратором.

Для обеспечения контроля целостности программной и информационной части МЭ «Z-2» `Tripwire` необходимо запускать с периодичностью как минимум раз в сутки, проверяя тем самым возможные изменения в критичных с точки зрения информационной безопасности файлах операционной системы и межсетевого экрана. Запуск может производится системным планировщиком `CRON`.

Результатом работы является список всех изменений в установленном наборе файлов, включая удаление и добавление файлов, со времени инсталляции или последнего обновления базы данных контрольных сумм, что позволяет гарантировать целостность программной и информационной части МЭ «Z-2».

11.1 Функционирование Tripwire

Tripwire использует два вида входных данных:

- описание файловой системы — объекта мониторинга;
- предварительно сгенерированную базу контрольных сумм текущей конфигурации.

Файл конфигурации содержит список файлов и/или каталогов с ассоциированными с ними списками атрибутов, подлежащих мониторингу. База контрольных сумм, генерируемая Tripwire содержит набор записей с именами файлов, значениями атрибутов, информацией о сигнатуре, избирательных масках и записях конфигурационного файла, на основе которого была сгенерирована база контрольных сумм.

11.1.1 Режимы функционирования

Tripwire функционирует в четырех режимах. Все программные операции выполняются в соответствии с данными конфигурационного файла — /opt/fw/tripwire/tw.conf.

Режим инициализации базы контрольных сумм предназначен для генерации первоначальной базы данных контрольных сумм, содержащую записи о каждом файле согласно спецификации файла конфигурации (tw.conf). Каждая запись в базе контрольных сумм содержит: имя файла, атрибуты, информацию о сигнатуре и запись о конфигурации, на которой была сгенерирована база контрольных сумм. Полученная база данных будет записана в директорию .databases, и после генерации должна быть перенесена администратором в директорию /opt/fw/tripwire. Эта операция осуществляется автоматически при первом запуске межсетевого экрана.

Режим проверки целостности предназначен для считывания данных из файла tw.conf и генерации новой базы контрольных сумм. Затем сравнивает вновь сгенерированную базу данных контрольных сумм с первоначальной, составляя список добавленных и удаленных файлов. Для всех измененных файлов определяется — нужно ли заносить информацию о произошедших событиях в создающийся отчет. Эта операция производится автоматически при запуске МЭ «Z-2».

Когда выявлены изменения файлов и они разрешены, то необходимо обновить первоначальную базу контрольных сумм. В Tripwire предусмотрено два режима обеспечения согласованности баз контрольных сумм:

Режим обновления базы контрольных сумм. В этом режиме Tripwire выдает список файлов или конфигурационных записей в командной строке. Записи базы контрольных сумм для этих файлов перегенерируются и новая база перезаписывается. Tripwire запрашивает у системного администратора разрешение на помещение базы контрольных сумм на безопасный носитель.

Режим интерактивного обновления базы контрольных сумм. В этом режиме Tripwire выводит список всех изменений, полученных в режиме проверки целостности. Для каждого изменения Tripwire спрашивает системного администратора, обновлять ли соответствующий файл или запись.

11.1.2 Расширяемость

Tripwire включает поддержку препроцессорного языка m4. Используя директивы @@include, @@ifdef, @@ifhost и @@define, системный администратор может создать файл-дескриптор, описывающий части файловой системы разделяемые на другие компьютеры. Tripwire может использоваться в системах из многих компьютеров. Файл конфигурации не обязательно держать на каждом из них, он может считываться с помощью файлов-дескрипторов и открываться во время вызова программы Tripwire. Tripwire не кодирует файл базы контрольных сумм, потому что в нем не содержится

информации, которая могла бы помочь взлому системы. Однако, интерфейс Tripwire поддерживает возможность кодирования другими средствами, если этого требует политика безопасности.

11.1.3 Сигнатуры

Tripwire позволяет изменять процедуру генерации сигнатуры и поддерживает более десяти сигнатур для каждого файла. В Tripwire включается реализации следующих методов: MD5, MD4, MD2 (RSA Data Security Message-Digest Algorithm), Snefru (Xerox Secure Hash Function) и SHA (NIST Secure Hash Function). Кроме того Tripwire содержит POSIX 1003.2 CRC-32 и CCITT CRC-16.

Поскольку каждая программа создания сигнатуры имеет свое собственное соотношение между производительностью и безопасностью, системный администратор может использовать ту, которая более полно отвечает требованиям безопасности. По умолчанию для проверки целостности используются сигнатуры MD5 и Snefru. Допускается определять различные типы сигнатур для каждого файла. Это дает системному администратору более широкие возможности.

В Tripwire включена программа siggen, которая генерирует сигнатуры для файлов, определенных в командной строке. Это удобный инструмент для создания всех доступных сигнатур для любых файлов.

11.1.4 Конфигурирование Tripwire

Конфигурирование Tripwire производится путем редактирования конфигурационного файла /opt/fw/triwire/tw.conf. Файл конфигурации tw.conf содержит списки файлов или каталогов для мониторинга и связанные с ними избирательные маски. Ниже приведен пример записи из файла tw.conf:

```
# file/dir      selection-mask
/etc            R       # all files under /etc
@@ifhost solaria.cs.purdue.edu
!/etc/lp               # except for SVR4 printer logs
@@endif
/etc/passwd     R+12    # you can't be too careful
/etc/mtab       L       # dynamic files
/etc/motb       L
/etc/utmp       L
=/var/tmp       R       # only the directory, not its contents
```

По умолчанию все файлы, содержащиеся в указанном каталоге, включаются в файл базы контрольных сумм при генерации.

Значениями маски могут являться следующие флаги:

i Номер inode
p Права доступа к файлу
n Количество ссылок на данный файл
u Идентификатор владельца файла
g Идентификатор группы владельца файла
s Размер файла
a Время доступа к файлу
m Время модификации файла
c Время создания или модификации inode

0 Использовать нулевую сигнатуру
1 Использовать сигнатуру MD5, RSA Data Security Message Digesting Algorithm
2 Использовать сигнатуру Snefru, Xerox Secure Hash Function
3 Использовать сигнатуру CRC-32, 32-bit Cyclic Redundancy Check
4 Использовать сигнатуру CRC-16, 16-bit Cyclic Redundancy Check
5 Использовать сигнатуру MD4, RSA Data Security Message Digesting Algorithm
6 Использовать сигнатуру MD2, RSA Data Security Message Digesting Algorithm
7 Использовать сигнатуру SHA, NIST Secure Hash Algorithm (NIST FIPS 180)
8 Использовать сигнатуру Haval

Символы «+» и «-» означают добавление и удаление указанных за ними флагов.

Например, избирательная маска может выглядеть так:

```
+pinugsm12-a
```

Tripwire интерпретирует эту запись следующим образом: «Занести в отчет изменения прав доступа к файлу, номеров inode, количества ссылок на этот файл, идентификатора владельца файла, идентификатора группы владельца файла, размера файла, времени модификации, при генерации базы данных использовать сигнатуры MD5 и Snefru. Игнорировать изменения времени доступа к файлу.»

Кроме вышеуказанных флагов, предусмотрены наборы шаблонов позволяющие системному администратору быстро классифицировать файлы по категориям:

R (read-only) — файлы только для чтения, игнорируется время доступа, при генерации базы данных используются сигнатуры MD5 и Snefru.
L (log file) — файлы статистики, игнорируется изменение размера файла, времени доступа и модификации и сигнатур.
> (growing log) — дописываемые файлы статистики, игнорируется времени доступа и модификации и сигнатур. Игнорируется увеличение размера файла.
N (ignore nothing) — не игнорировать никакие данные.
E (ignore everything) — игнорировать все данные.

Если какой-либо атрибут установленный для мониторинга был изменен, то в отчете записывается как ожидаемое, так и актуальное значение атрибута. Ниже приведен пример отчета об изменениях атрибута времени последней модификации файла:

```
changed: -rw-r -r-  root       20 Sep 17 13:46:43 1993 /.rhosts
### Attr   Observed (what it is)     Expected (what it should be)
### ====== ========================= ============================
/.rhosts
st_mtime:  Fri Sep 17 13:46:43 2001  Tue Sep 14 20:05:10 2001
```

11.2 Проверка целостности

Для выполнения проверки целостности необходимо произвести следующие действия:

1. Если база контрольных сумм не была до этого создана, ее необходимо создать. Для этого необходимо запустить `tripwire` в режиме инициализации базы контрольных сумм:

   ```
   # /opt/fw/bin/tripwire -initialize
   ```

2. Выполнить команду для проверки целостности:

   ```
   # /opt/fw/bin/tripwire
   ```

3. В случае успешной проверки `Tripwire` выдаст следующие результаты:

   ```
   ### Phase 1:   Reading configuration file
   ### Phase 2:   Generating file list
   ### Phase 3:   Creating file information database
   ### Phase 4:   Searching for inconsistencies
   ###
   ###                     Total files scanned:            15798
   ###                           Files added:              0
   ###                           Files deleted:            0
   ###                           Files changed:            15720
   ###
   ###                     After applying rules:
   ###                           Changes discarded:        15720
   ###                           Changes remaining:        0
   ###
   ```

   Количество проверенных файлов может отличаться от вышеприведенного примера и определяется в конфигурационном файле `/opt/fw/tripwire/tw.conf`. Если файлы или директории, подлежащие мониторингу были изменены, то в отчете будут приведены имена измененных файлов или директорий, а также будет указано, какие именно атрибуты указанных файлов подверглись изменениям, например:

   ```
   changed: -rw-r -r-  root       20 Sep 17 13:46:43 1993 /.rhosts
   ### Attr   Observed (what it is)     Expected (what it should be)
   ### ====== ========================= ============================
   /.rhosts
   st_mtime:  Fri Sep 17 13:46:43 2001  Tue Sep 14 20:05:10 2001
   ```

Глава 12

Утилиты командной строки

12.1 Утилиты для сохранения и восстановления конфигурации МЭ

В состав МЭ «Z-2» входят утилиты для сохранения и восстановления конфигурации МЭ, предназначенные для восстановления свойств МЭ после возможных сбоев или отказов оборудования, а также после повреждения или непреднамеренного изменения конфигурационных файлов МЭ. [1]

12.1.1 Утилита **save_config.sh**

12.1.1.1 Общее описание

Утилита `save_config.sh` предназначена для сохранения текущей конфигурации МЭ. `save_config.sh` расположена в директориии `/opt/fw/sbin`.

Данная утилита сохраняет конфигурационные файлы МЭ на устройстве, определенном администратором межсетевого экрана. Список файлов и директорий, подлежащих сохранению определяется в файле `/opt/fw/etc/savelist` и может быть расширен администратором по своему усмотрению.

12.1.1.2 Пример использования утилиты **save_config.sh**

```
# /opt/fw/sbin/save_config.sh
Z-2 configuration backup starting ...
Enter save device (default is /dev/rfd0) : /var/tmp/backup.tar
```

После запуска утилита `save_config.sh` спрашивает имя устройства, предназначенного для записи конфигурационных файлов. Это может быть ленточный накопитель, дискета, файл и т.д., по умолчанию используется устройство /dev/rfd0, производящее запись на дискету. В данном примере в качестве устройства для сохранения используется файл `/var/tmp/backup.tar.`.

После определения имени устройства утилита спрашивает администратора о готовности, и в случае получения положительного ответа производит сохранение конфигурации МЭ :

```
Using /var/tmp/backup device as save device
Make shure that you have write permissions on /var/tmp/backup
```

[1] Описываемые ниже утилиты предназначены только для сохранения и восстановления конфигурационных файлов МЭ. В случае критических сбоев, повлекших за собой нарушения или невозможность функционирования ОС, восстановление работоспособности операционной системы должно проводиться штатными системными средствами.

```
Ready to start backup procedure. Continue ? (y/n) : y
a /etc/inet/hosts 1K
a /etc/pam.conf 3K
a /etc/resolv.conf 1K
a /etc/syslog.conf 2K
a /etc/mail/sendmail.cf 35K
a /etc/mail/aliases 2K
a /opt/fw/etc/ 0K
a /opt/fw/etc/firewall.xml 13K
.
.
.
a /opt/fw/var/skeykeys 0K
a /opt/fw/var/pamdb.db.pag 0K
a /opt/fw/var/pamdb.db.dir 0K
Z-2 configuration backup done.
```

12.1.2 Утилита restore_config.sh

12.1.2.1 Общее описание

Утилита restore_config.sh предназначена для восстановления сохраненной конфигурации МЭ и расположена в директориии /opt/fw/sbin.

Данная утилита восстанавливает конфигурационные файлы МЭ, ранее сохраненные с помощью утилиты save_config.sh с устройства, определенного администратором. Для корректной работы данная утилита требует права суперпользователя. После восстановления конфигурации межсетевой экран следует перезагрузить.

12.1.2.2 Пример использования утилиты restore_config.sh

```
# /opt/fw/sbin/restore_config.sh
Z-2 configuration restore starting ...
Enter restore device (default is /dev/rfd0) : /var/tmp/backup.tar
```

После запуска утилита restore_config.sh спрашивает имя устройства, на котором находится сохраненная конфигурация. По умолчанию используется устройство /dev/rfd0, производящее чтение с дискеты. В данном примере в качестве устройства с сохраненной конфигурационной информацией используется файл /var/tmp/backup.tar:

```
Using /var/tmp/backup.tar device as restore device

Ready to start restore procedure. Continue ? (y/n) : y
x /etc/inet/hosts, 252 bytes, 1 tape blocks
x /etc/pam.conf, 2503 bytes, 5 tape blocks
x /etc/resolv.conf, 52 bytes, 1 tape blocks
x /etc/syslog.conf, 1054 bytes, 3 tape blocks
x /etc/mail/sendmail.cf, 35625 bytes, 70 tape blocks
x /etc/mail/aliases, 1348 bytes, 3 tape blocks
x /opt/fw/etc, 0 bytes, 0 tape blocks
x /opt/fw/etc/firewall.xml, 13004 bytes, 26 tape blocks
```

```
.
.
.
x /opt/fw/var/skeykeys, 0 bytes, 0 tape blocks
x /opt/fw/var/pamdb.db.pag, 0 bytes, 0 tape blocks
x /opt/fw/var/pamdb.db.dir, 0 bytes, 0 tape blocks
Z-2 configuration restore done.
```

12.2 Программа архивирования протоколов logrotate

Программа `logrotate` предназначена для архивирования и ротации сохраненных копий системных протоколов. По-умолчанию, программа архивирует файлы системных сообщений */var/adm/messages*. Сохраненные копии протоколов помещаются в каталог */var/adm/OLD*.

В процессе архивирования, программа `logrotate` выполняет следующие действия:

1. Копирует файл */var/adm/messages* в каталог */var/adm/OLD*;
2. Файлы */var/adm/OLD/messages*.N.* переименовываются по правилу:
   messages.N.* -> messages.(N+1).*,
   где N - целое число, N = 1. . . 10.
3. Переименовывает файл */var/adm/OLD/messages* в */var/adm/OLD/messages.0*.
4. Архивирует файл */var/adm/OLD/messages.0* с помощью программы *gzip*.
5. Удаляет содержимое файла */var/adm/messages*.

Программа `logrotate` не использует аргументы командной строки. Запуск программы `logrotate` осуществляется из командной строки:

```
/opt/fw/bin/logrotate
```

Пользователь, запускающий программу `logrotate`, должен иметь доступ по записи в каталоги */var/adm* и */var/adm/OLD*.

Примечание При инсталляции межсетевого экрана в планировщике задач создается запись, запускающая программу `logrotate` в 2 часа ночи.

12.3 Утилита для сбора информации об установленных пакетах в системе

Для сбора информации об установленных пакетах в системе предназначены утилиты `report.sh` и `report-z2.sh`. В процессе работы утилита `report.sh` выдает информацию о конфигурации ОС, обо всех установленных пакетах МЭ Z-2 и ККМП Тропа (дата установки, версия, содержимое конфигурационных файлов и пр.). Утилита `report-z2.sh` выдает информацию только о МЭ Z-2.

Эта информация позволяет наиболее точно диагностировать проблемы в работе продуктов.

Утилиты запускаются из командной строки без опций:

```
# sh /opt/fw/bin/report.sh
```

```
# sh /opt/fw/bin/report-z2.sh
```

Глава 13

Проверка на вирусы с использованием Symantec ScanEngine

Прикладные шлюзы протоколов SMTP, FTP и FTP-модуль протокола HTTP содержат поддержку проверки передаваемых данных на вирусы с использованием антивирусного сервера Symantec ScanEngine.

Прикладной шлюз протокола SMTP осуществляет проверку сообщений в программе почтового ретранслятора непосредственно перед отправкой сообщения. В случае обнаружения вируса антивирус ScanEngine осуществляет исправление передаваемого письма и добавляет в почтовое сообщение подробную информацию о проделанных действиях. Антивирус самостоятельно осуществляет проверку прикрепленных к письмам файлов и декомпрессию основных типов архивный файлов. Подробную информацию об этом можно получить в документации на ScanEngine.

Прикладной шлюз протокола FTP и FTP-модуль шлюза протокола HTTP в режиме антивирусной проверки сохраняет передаваемый файл на сервере МЭ, осуществляет его проверку и после ее окончания передает файл пользователю.

13.1 Инсталляция Symantec ScanEngine

`Symantec ScanEngine` может быть установлен на сервера Sparc/Solaris, Intel/Linux и Windows/Linux. Антивирус может быть установлен на сервер МЭ, но в случае большой нагрузки рекомендуется установка антивируса на выделенный сервер. Подробно процедура инсталляции и настройки ScanEngine приведена в его руководстве по инсталляции.

Для использования ScanEngine следует выбрать следующие настройки:

- выбрать `ICAP` протокол работы `ScanEngine`;
- использовать порт по-умолчанию: 1344;
- разрешить антивирусу модифицировать почтовые сообщения.

Глава 14

Глоссарий

Алиас (в графическом интерфейсе) именованное определение какого-либо объекта, используемого в настройках. Например: *алиас хоста* определяет IP адрес хоста, *алиас сети* определяет IP-адрес и маску сети. Для всех управляемых серверов используется единый набор алиасов, хранящийся на *сервере алиасов* — специально выбранном для этого сервере (см. книгу «*«Универсальный графический интерфейс 2.4 Руководство администратора»*»).

Алиас (в программе sendmail) определяет псевдоним почтового адреса, либо определяет адрес рассылки для группы почтовых адресов.

Доверенный центр сертификации центр сертификации ключей, используемый для проверки достоверности сертификата открытого ключа.

Закрытый ключ конфиденциальная часть *ключевой пары*.

Ключевая пара состоит из закрытого ключа и открытого ключа. Закрытый ключ является конфиденциальным и используется только для доказательства владения открытым ключом. Открытый ключ не является конфиденциальным и распространяется без ограничений. См также *сертификат открытого ключа*.

Определяющее имя уникальное имя владельца ключевой пары. Записывается в формате X.500. В МЭ «Z-2» определяющее имя администратора является его регистрационным именем.

Открытый ключ не конфиденциальная часть *ключевой пары*.

Пакетный фильтр компонент МЭ «Z-2», осуществляющий:

- фильтрацию пакетов на транспортном и сетевом уровнях
- контроль установленных через МЭ соединений
- трансляцию сетевых адресов
- учет трафика на сетевом уровне
- поддержку *прозрачного* режима работы прикладных шлюзов

Прикладной шлюз компонент МЭ «Z-2», осуществляющий ретрансляцию запросов пользователя. Схема функционирования прикладных шлюзов такова: шлюз получает запрос пользователя и от своего имени осуществляет его. Прикладные шлюзы делятся на две категории: специализированных прикладные шлюзы (например шлюз HTTP) и шлюзы общего назначения (универсальный прикладной шлюз TCP и универсальный прикладной шлюз UDP). Специализированные шлюзы осуществляют контроль прикладных протоколов и фильтрацию на основе адресов прикладного уровня.

Псевдосессия протокола UDP представление механизма двухстороннего обмена по протоколу `UDP`. Сессия создается при поступлении первого пакета. Ответный пакет считается принадлежащим в сессии, если его адрес источника назначения соответствует адресу назначения исходного пакета и порт назначения соответствует порту источника исходного пакета. Сессия считается закрытой по истечении определенного времени с момента получения последнего пакета в сессии. Таким механизмом описываются многие (но не все) схемы обмена по протоколу `UDP`. Псевдосессии поддерживаются пакетным фильтром и прикладным шлюзом `gudp-gw`.

Прозрачный режим работы прикладного шлюза режим работы прикладного шлюза, при котором

пользователю не требуется осуществлять дополнительных настроек ПО. Вместо этого запросы пользователя, направленные через МЭ принудительно перенаправляются в прикладной шлюз пакетным фильтром.

Сервер аутентификации обрабатывает запросы на аутентификацию пользователей от прикладных шлюзов. Обеспечивает работу прикладных шлюзов с аутентификационными модулями сторонних производителей по PAM протоколу.

Сертификат открытого ключа содержит *открытый ключ*, определяющее имя владельца сертификата и срок его действия. Сертификат может быть подписан *доверенным центром сертификации*. Сертификат открытого ключа не содержит конфиденциальной информации.

Сетевая трансляция адресов трансляция адреса отправителя или получателя пакета. Осуществляется пакетным фильтром МЭ «Z-2». Трансляция адресов применяется для сокрытия внутренней структуры сети, организации работы прозрачных шлюзов, исправления ошибок проектирования сети.

Спам непрошенная почтовая корреспонденция рекламного характера.

Управляющий сервер компонент МЭ «Z-2», предоставляющий модули управления сервером для графического интерфейса.

Шлюз См. *прикладной шлюз* и *ядерный шлюз*.

Ядерный шлюз компонент пакетного фильтра МЭ «Z-2». Шлюзы ядерного уровня предназначены только для пропускания определенных прикладных протоколов через МЭ «Z-2». Они не осуществляют фильтрацию и контроль за протоколом прикладного уровня.

Глава 15

Часто задаваемые вопросы

A Прикладные шлюзы

Хочу проверить работу прикладного шлюза HTTP в прозрачном режиме. Все настроил, проверяю `telnet`'ом и получаю ошибку: `No host header in request` Даже если вы работаете в прозрачном режиме, необходимо указывать Host-заголовок. Для HTTP 1.0 это требование *de-facto*, для HTTP 1.1 — описано в стандарте. Без host заголовка вы в любом случае не сможете получать данные с хостов с виртуальным хостингом, которые составляют значительный процент хостов в Интернет. Все современные броузеры и другие HTTP-клиенты передают Host-заголовок даже в HTTP 1.0.

Как настроить обслуживание `https` в прозрачном режиме. Нужно ли его заворачивать на шлюз HTTP? Необходимо пропустить https-трафика (443/tcp) через пакетный фильтр наружу, либо перенаправлять его на универсальный прикладной шлюз TCP. Перенаправлять https-трафик на прикладной шлюз HTTP в прозрачном режиме не следует.

Как мне запретить доступ к шлюзу HTTP определенным пользователям так, чтобы им вместо обрыва соединения выдавалась страничка со словами Permission Denied? Нужно добавить правило, в которое бы попадал этот IP-адрес и не добавлять в него outbound правил вообще — тогда соединение с пользователем будет установлено, но на любой его запрос будет ответ об отсутствии доступа.

Почему smtp-ретранслятор не хочет отправлять написанные мной тестовые письма, посланные мною с помощью telnet на шлюз SMTP? По RFC822, в письмо должно состоять из заголовка и тела, разделенных пустой строкой. Письма без подобного разделения считаются некорректными. Пишите тестовые письма с заголовками, например:

```
MAIL FROM: bofh@hack.ru
RCPT TO: user@mail.ru
DATA
X-Mailer: BOFH

test mail from bofh. Beware !
.
QUIT
```

Как перевести прикладной шлюз в отладочный режим ? Отладочные сообщения прикладных шлюзов пишутся в `local0.debug`. По умолчанию syslog не сохраняет эти сообщения. Настройте ваш syslog чтобы он писал эти сообщения.

После настройки syslog можно переводить прикладной шлюз в отладочный режим. Для этого нужно послать сигнал USR1 всем процессам прикладного шлюза, например так:

```
# pkill -USR1 http_gw
```

Примечание при нормальном функционировании прикладного шлюз включать отладочный режим не рекомендуется из-за большого количества выводимой отладочной информации

Б IP–Filter

Я настроил Пакетный фильтр таким образом, чтобы он пропускал только необходимые мне протоколы. После этого при соединении с ftp, telnet или smtp серверами возникают задержки порядка 30 секунд. DNS настроен корректно. В чем причина задержек ? Вышеупомянутые сервера как правило используют службу ident для определения идентификатора пользователя, устанавливающего соединение. Для этого на IP-адрес клиента сервер устанавливает соединение по TCP-порту 113. Так как этот порт на межсетевом экране заблокирован, то сервер пытается соединится в течении определенного промежутка времени, что и приводит к задержкам. Чтобы от них избавится, в настройках Пакетного фильтра для порта 113 должно быть правило с действием reject, а не deny.

Нужно ли добавлять правила для разрешения FTP-DATA соединений от клиентов и серверов в правила Пакетного фильтра? Нет, этого делать не нужно — прикладной шлюз FTP сам создает необходимые для работы правила.

Почему я не могу задать правила Пакетного фильтра или NAT для loopback интерфейса ? loopback интерфейс в Solaris является «поддельным» и не может использоваться для IP-фильтрации, NAT, snoop, etc.

После установки и настройки Z-2 система некоторое время работает нормально, но потом резко возрастают задержки. В чем проблема? Вероятнее всего у вас переполняется таблица состояний или таблица NAT в Пакетном фильтре. По умолчанию размер таблицы ограничен 4000 записями и этого недостаточно при работе больших сетей через МЭ. Выяснить это можно при помощи команды ipfstat -s:

```
root@fw:/[113]# ipfstat -s
IP states added:
951727 TCP
19067 UDP
10 ICMP
70743826 hits
979757 misses
0 maximum
0 no memory
7259 bkts in use
8851 active
19069 expired
942884 closed
```

Необходимо обратить внимание на поле 'maximum' и 'no memory' Число в поле 'maximum' — сколько раз новая сессия не была добавлена из-за переполнения таблицы состояний, число в поле 'no memory' — сколько раз новая сессия не была добавлена из-за нехватки памяти в системе. Если значение поля 'maximum' отличается от нуля, необходимо увеличить размер таблицы состояний. Сделать это можно в файле /etc/system :

```
* ipf: adjust the state table sizes so we have enough buckets.
* IPSTATE_MAX (=fr_statemax) should be ~70% of IPSTATE_SIZE
* IPSTATE_SIZE (=fr_statesize) has to be a prime number
set ipf:fr_statemax = 14000
set ipf:fr_statesize = 20021
```

Размер таблицы состояний (fr_statesize) должен быть простым числом, а максимально допустимое число состояний (fr_statemax) должно быть приблизительно равно 70% от fr_statesize (Списокк простых чисел может быть найден на http://www.utm.edu/research/primes/). Кроме этого, можно уменьшить тайм-аут на неактивные tcp-сессии (по умолчанию он равен 5 суток):

```
* ipf: adjust the default tcp timeouts downward so that
* idle (dead) and half closed states get killed off quicker.
```

```
set ipf:fr_tcpidletimeout = 172800
set ipf:fr_tcphalfclosed = 7200
```

Размер таблица для NAT устанавливается параметром ipf_nattable_sz :

```
* ipf: ajust NATTABLESIZE
set ipf:ipf_nattable_sz = 20021
```

Статистика для NAT выводится командой ipnat -s

Я увеличил размер таблицы состояний, но ее все равно не хватает. Почему? Убедитесь, что во всех правилах для TCP с keep state имеется запись flags S. В противном случае состояние заводится не только при начале TCP-сессии, но и для любого пакета этой сессии, пришедшего вне своей очереди. Это приводит к неограниченному росту числа записей в таблице состояний.

B GUI

При работе с GUI в X'ах вылезают пустые или полупрорисованные окошки, что делать? Это происходит из-за ошибок в оконном менеджере fwvm. Используете какой-нибудь другой менеджер, например Sawfish или CDE.

Медленно работает GUI под Windows, почему ?

- Необходимо как минимум 128 Mb памяти
- Если на машину установлен антивирус AVP/DrWeb или что-то подобное, отключите режим постоянной проверки памяти на вирусы
- Убедитесь что у вас разрешается IP-адрес межсетевого экрана с машины, на которой запускается GUI. В противном случае GUI будет работать _очень_ медленно. JFY, в Windows есть аналог /etc/hosts

GUI не соединяется к МЭ и выдает ошибку: `Connect attempt failed. Make sure remote server is up`. Что делать?

- Убедитесь что на МЭ запущен managment server
- Убедитесь что с машины, на которой запускается GUI разрешен доступ к МЭ на порт 4040 (managment server listen port)
- Убедитесь в том, что версия Java, установленная на машине с GUI соответствует версии Java, поставляемой на диске с Z-2. Узнать версию Java можно командой java -version

Г Solairs

Как узнать, в каком режиме работает Solaris — 32-х или 64-х битном? Введите команду *isainfo -b*

Как сделать так, чтобы Solaris загружался в 64-х битном режиме? От имени суперпользователя введите команду

```
eeprom boot-file=kernel/sparcv9/unix
```

, после чего перезагрузите компьютер

Примечание 64-х битный режим доступен только на платформе SPARC

Д Прочее

Как узнать версию установленного Z-2? cat /opt/fw/etc/release . Кроме того, прикладные шлюзы сообщают их версию при запуске с ключом '-v', так что если у вас ставились какие-либо программные коррекции, после инсталляции вы можете узнать реальные версии.

После установки Z-2 и настройки Пакетного фильтра система не работает или работает с большими задержками, что делать? Это происходит если у вас в системе присутствует интерфейс eri или gigabit ethernet (например, сервер Sun Blade). Эти интерфейсы по умолчанию реализуют hardware checksumming, что приводит к конфликту с модулем IP-Filter. Чтобы отключить hardwire checksumming, необходимо в файл /etc/system добавить строчку:

```
* hardware checksum breaks ip-filter, need to turn it off
set ip:dohwcksum = 0
```

после чего перезагрузить Solaris

Сколько Z-2 занимает места на диске? Примерно 220Mb в /opt/fw, 6Mb — GUI, еще примерно 2Mb — Пакетный фильтр.

При инсталляции МЭ программа инсталляции спрашивает *Please enter GUI manager IP address for this firewall*. Что это за адрес, как он используется и как его можно изменить после инсталляции МЭ ? Это адрес машины, с которой будет разрешено управление МЭ с помощью графического интерфейса. Когда Вы вводите этот адрес, программа инсталляции создает правило пакетного фильтра, разрешающее доступ с указанного адреса на МЭ порт 4040. Это сделано для того, чтобы при изменении политики безопасности «по умолчанию» управляющая сессия не была прервана перезагрузкой правил фильтрации. Созданное правило действует только до первого применения правил с графического интерфейса, поэтому изменять его не имеет смысла. Через GUI подобное правило задается в разделе *Пакетный фильтр*, пункт *Разрешать присоединяться к управляющему серверу с*