МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
ЧЕРНИГОВСКИЙ НАЦИОНАЛЬНЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ
КАФЕДРА ИНФОРМАЦИОННЫХ И КОМПЬЮТЕРНЫХ СИСТЕМ

ПРОЕКТИРОВАНИЕ КОМПЬЮТЕРНЫХ СИСТЕМ И СЕТЕЙ

Методические указания
к выполнению самостоятельных работ по дисциплине „Проектирование компьютерных систем и сетей”
для студентов специальностей
7.05010201 “Компьютерные системы и сети”
7.05010202 “Системное программирование”
7.05010203 “Специализированные компьютерные системы”
8.05010201 “Компьютерные системы и сети”
8.05010202 “Системное программирование”
8.05010203 “Специализированные компьютерные системы”

Утверждено
на заседании кафедры
информационных и компьютерных систем

Протокол № 2 от «27» сентября 2013 г.

Чернигов ЧНТУ 2013

Проектування комп’ютерних систем та мереж. Методичні вказівки до виконання самостійних робіт з дисципліни “Проектування комп’ютерних систем та мереж” для студентів спеціальностей 7.05010201 “Комп’ютерні системи та мережі”, 7.05010202 “Системне програмування”, 7.05010203 “Спеціалізовані комп’ютерні системи”, 8.05010201 “Комп’ютерні системи та мережі”, 8.05010202 “Системне програмування”, 8.05010203 “Спеціалізовані комп’ютерні системи” / Укл. Є. В. Риндич. – Чернігів: ЧНТУ, 2013. – 115 с., рос. мовою.

Укладачі: Риндич Євген Володимирович, кандидат технічних наук, доцент кафедри інформаційних та комп’ютерних систем

Відповідальний за випуск: Казимир Володимир Вікторович, завідувач кафедри інформаційних та комп’ютерних систем, доктор технічних наук, професор

Рецензент: Нікітенко Євгеній Васильович, кандидат фізико-математичних наук, доцент кафедри інформаційних та комп’ютерних систем Чернігівського національного технологічного університету

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 5
1 Самостоятельная работа №1. Создание виртуальной частной сети на базе PPTP сервера POPTOP 7
1.1 Цель работы 7
1.2 Теоретические сведения 7
1.3 Ход работы 8
1.3.1 Установка необходимых пакетов для создания виртуальной частной сети 8
1.3.2 Загрузка пакетов 9
1.3.3 Распаковка пакетов 9
1.3.4 Установка пакета ppp 9
1.3.5 Установка пакета pptpd 9
1.3.6 Создание конфигурационных файлов pptp 10
1.3.7 Добавление клиента виртуальной частной сети 13
1.3.8 Тестирование работы сервера pptp 14
1.3.9 Настройка клиента созданной виртуальной частной сети из ОС Windows 14
1.3.10 Настройка клиента созданной виртуальной частной сети из ОС Linux 14
1.3.11 Включение форвардинга пакетов 16
1.3.12 Создание частной приватной сети 16
1.4 Содержание отчета 17
1.5 Контрольные вопросы 17
2 Самостоятельная работа №2. Настройка аутентификации, авторизации и учёта VPN пользователей посредством RADIUS сервера FreeRADIUS, с применением модуля rlm_sql 18
2.1 Цель работы 18
2.2 Теоретические сведения 18
2.3 Ход работы 24
2.3.1 Установка необходимых пакетов 24
2.3.2 Загрузка пакетов 24
2.3.3 Распаковка пакетов 25
2.3.4 Установка пакета postgresql 25
2.3.5 Установка пакета freeradius 25
2.3.6 Создание БД для модуля rlm_sql/rlm_sql_postresql сервера FreeRADIUS 26
2.3.7 Добавление пользователей в созданную БД 27
2.3.8 Конфигурирование сервера FreeRADIUS 27
2.3.9 Тестирование работы сервера FreeRADIUS 51
2.3.10 Конфигурирование pptp сервера poptop 53
2.3.11 Проверка работы vpn клиента из ОС Windows и ОС Linux 57
2.4 Содержание отчета 60
2.5 Контрольные вопросы 60
3 Самостоятельная работа №3. Настройка почтового сервера postfix 62
3.1 Цель работы 62
3.2 Краткие теоретические сведения 62
3.3 Ход работы 63
3.3.1 Установка необходимых пакетов 63
3.3.2 Установка пакета postfix 63
3.3.3 Настройка postfix 63
3.3.4 Проверка заданных настроек postfix 74
3.3.5 Установка и настройка спамфильтра 75
3.3.6 Интеграция спамфильтра в postfix 76
3.3.7 Интеграция антивируса в postfix 79

3.4 Содержание отчета 94
3.5 Контрольные вопросы 94
4 Самостоятельная работа №4. Преобразование сетевых адресов и статистика в фильтрах iptables. Типовые решения защиты сетей 95
4.1 Цель работы 95
4.2 Краткие теоретические сведения 95
4.3 Ход работы 97
4.3.1 Установка необходимых пакетов 97
4.3.2 Преобразование сетевых адресов 98
4.3.3 Преобразование адреса при помощи SNAT 98
4.3.4 Преобразование адреса назначения пакета (Destination Network Adress Translation) 99
4.3.5 Преобразование адреса при помощи MASQUERADE 99
4.3.6 Реализация подсчета трафика с помощью iptables на локальном хосте 99
4.3.7 Конфигурирование типового сетевого экрана для домашней локальной сети с выходом в интернет через PPTP подключение 100
4.4 Содержание отчета 102
4.5 Контрольные вопросы 102
5 Самостоятельная работа №5. Настройка HTTP-прокси сервера Squid 104
5.1 Цель работы 104
5.2 Краткие теоретические сведения 104
5.3 Ход работы 105
5.3.1 Установка необходимых пакетов 105
5.3.2 Загрузка пакетов 106
5.3.3 Распаковка пакетов 106
5.3.4 Установка пакета squid 106
5.3.5 Настройка проксирования HTTP трафика 106
5.3.6 Фильтрация по URL 108
5.3.7 Фильтрация по ключевым словам в URL 109
5.3.8 RADIUS авторизация 110
5.3.9 Ограничение пропускной способности 112
5.4 Содержание отчета 113
5.5 Контрольные вопросы 113
Рекомендованная литература 115

ВВЕДЕНИЕ

Редкий серьезный деловой человек, профессиональный программист или системный оператор не может представить себе полноценную работу без использования такого мощного, оперативного и удобного сочетания как обычная телефонная линия, модем и компьютерная сеть. В то время как первые две составляющие всего лишь техническая сторона новой организации информационного обмена между пользователями, компьютерная сеть - это та глобальная идея, объединяющая разрозненных обладателей компьютеров и модемов, систематизирующая и управляющая хаотически предъявляемыми требованиями и запросами по быстрому информационному обслуживанию, моментальной обработкой коммерческих предложений, услугами личной конфиденциальной переписки и т.д. и т.п.

Сейчас, в условиях многократно возрастающих каждый год информационных потоков, уже практически невозможно вообразить четкое взаимодействие банковских структур, торговых и посреднических фирм, государственных учреждений и других организаций без современной вычислительной техники и компьютерных сетей. В противном случае пришлось бы содержать гигантский штат обработчиков бумажных документов и курьеров, причем надежность и быстрота функционирования такой системы все равно была бы значительно ниже предоставляемой модемной связью и компьютерными сетями. А ведь каждая минута задержки в пересылке важных информационных сообщений может вылиться в весьма ощутимые денежные потери и имиджевые крахи.

Результатом эволюции компьютерных технологий явились вычислительные сети. Вычислительная сеть – это сложный комплекс взаимосвязанных и согласованно функционирующих программных и аппаратных компонентов.

Комплекс аппаратно – программных средств сети может быть описан многоуровневой моделью.

В основе любой сети лежит аппаратный слой, который включает компьютеры различных классов. Набор компьютеров в сети должен соответствовать набору разнообразных задач, решаемых сетью.

Второй слой составляет разнообразное сетевое оборудование, необходимое для создания локально-вычислительных сетей, и коммуникационное оборудование для связи с глобальными сетями. Коммуникационные устройства играют не менее важную роль, чем компьютеры, которые являются основными элементами по обработке данных.

Третьим слоем являются операционные системы, которые составляют программную основу сети. При построении сетевой структуры важно учитывать насколько эффективно данная операционная система может взаимодействовать с другими операционными системами сети, насколько она способна обеспечить безопасность и защиту данных и т. д.

Самым верхним слоем сетевых средств являются различные сетевые приложения, такие как сетевые базы данных, почтовые системы, средства архивирования данных и др. Важно знать совместимость различных сетевых приложений.

В настоящее время использование вычислительных сетей даёт предприятию многочисленные возможности. Конечной целью использования вычислительных сетей на предприятии является повышение эффективности его работы, которое может выражаться, например, в увеличении прибыли предприятия. Если же рассматривать вопрос внедрения ЛВС в работу учреждений (с учётом появления новых возможностей у предприятия) более глубоко, то из этого вытекают ещё несколько преимуществ.

Концептуальным преимуществом распределённых систем и, следовательно, сетей перед централизованными системами является их способность выполнять параллельные вычисления, что увеличивает производительность. Такие системы имеют лучшее соотношение производительность – стоимость, чем централизованные системы.

Следующее преимущество – это совместное использование пользователями данных и устройств: цветных принтеров, графопостроителей, модемов, оптических дисков.

В последнее время стал преобладать другой побудительный мотив развертывания сетей, гораздо более важный, чем экономия средств при разделении дорогостоящих ресурсов. Этим мотивом стало стремление обеспечить пользователям сети оперативный доступ к обширной корпоративной информации.

Использование сети приводит к совершенствованию коммуникаций, т.е. к улучшению процесса обмена информацией и взаимодействия между сотрудниками предприятия, а также его клиентами и поставщиками. Сети снижают потребность предприятий в других формах передачи информации, таких как телефон или обычная почта. Зачастую вычислительные сети на предприятии развёртываются из-за возможности организации электронной почты.

Безусловно, вычислительные сети имеют и свои проблемы (сложности с совместимостью программного обеспечения, проблемы с транспортировкой сообщений по каналам связи с учётом обеспечения надежности и производительности), но главным доказательством эффективности является бесспорный факт их повсеместного распространения. Всё больше и больше появляются крупные сети с сотнями рабочих станций и десятками серверов.

1 Самостоятельная работа №1. Создание виртуальной частной сети на базе PPTP сервера POPTOP

1.1 Цель работы

Установка и конфигурирование сервера POPTOP под операционной системой Linux. Конфигурирование клиентов виртуальной частной сети под операционными системами Linux и Windows.

1.2 Теоретические сведения

Виртуальная частная сеть (Virtual Private Network – VPN) – логическая сеть, создаваемая поверх другой сети, например интернет. Несмотря на то, что коммуникации осуществляются по публичным сетям, с использованием небезопасных протоколов, за счёт шифрования создаются закрытые от посторонних каналы обмена информацией.

Чаще всего для создания виртуальной сети используется инкапсуляция протокола PPP (Point-to-Point Protocol – протокол двухточечного соединения RFC1331), который изначально был создан для коммуникации линий, в какой-нибудь другой протокол. Из наиболее распространенных можно отметить PPTP (Point-to-Point Tunneling Protocol) – GRE-инкапсуляцию (Generic Routing Encapsulation – общая инкапсуляция маршрутов) PPP через существующую TCP/IP-сеть, и PPPoE (Point-to-Point Protocol over Ethernet) – инкапсуляцию PPP в кадры Ethernet. Также существуют другие протоколы предоставляющие возможность формирования защищенных каналов (IPSec, SSH, ViPNet и др.).

Протокол PPP состоит из двух частей. Первая это механизмы фрагментирования и декодирования пакетов, вторая это группа протоколов именуемых LCP (Link Control Protocol), IPCP (Internet Protocol Control Protocol), PAP (Password Authentication Protocol) и CHAP (Challenge Handshake Authentication Protocol) и др. для согласования настроек соединения и для идентификации.

PAP протокол – это протокол простой проверки подлинности, предусматривающий отправку имени пользователя и пароля на сервер удаленного доступа открытым текстом (без шифрования). Протокол PAP крайне ненадежен, поскольку пересылаемые пароли можно легко читать в пакетах PPP, которыми обмениваются стороны в ходе проверки подлинности. Обычно PAP используется только при подключении к старым серверам удаленного доступа на базе UNIX, которые не поддерживают никакие другие протоколы проверки подлинности.

CHAP протокол основан на широко распространенном алгоритме проверки подлинности, предусматривающем передачу не самого пароля пользователя, а косвенных сведений о нем. При использовании CHAP сервер удаленного доступа отправляет клиенту строку запроса. На основе этой строки и пароля пользователя клиент удаленного доступа вычисляет хеш-код MD5 (Message Digest-5). Хеш-функция является алгоритмом одностороннего

(необратимого) шифрования, поскольку значение хеш-функции для блока данных вычислить легко, а определить исходный блок по хеш-коду с математической точки зрения невозможно. Хеш-код MD5 передается серверу удаленного доступа. Сервер, которому доступен пароль пользователя, выполняет те же самые вычисления и сравнивает результат с хеш-кодом, полученным от клиента. В случае совпадения учетные данные клиента удаленного доступа считаются подлинными.

В операционной системе Linux сервером PPTP выступает POPTOP, распространяемый по лицензии GPL. POPTOP сам всего лишь инкапсулирует PPP в GRE-соединение. Для создания PPP-соединения он использует pppd. В качестве PPPoE-сервера может выступать rp-pppoe. Как и POPTOP, rp-pppoe использует pppd для создания ppp-соединений. Для BSD существует еще несколько реализаций PPTP- и PPPoE-серверов, в частности mpd и pppoed. Они имеют свои плюсы и минусы по сравнению с POPTOP и rp-pppoe.

Пакет ppp состоит из нескольких частей:

– код ядра (уже включён в ядра старше 2.2) компилируемый или в само ядро или в модуль ядра, который создаёт сетевой интерфейс и производит обмен пакетами между последовательным портом, сетевой частью ОС и демоном PPP (pppd);
– демон PPP (pppd), который взаимодействует со стороной устанавливающей соединение и настраивает сетевые интерфейсы ppp. Pppd включает поддержку идентификации, таким образом возможно производить контроль кто может создавать PPP соединение и какой IP адрес можно использовать;
– дополнительные модули (плагины) демона PPP.

Пакет pptpd состоит из нескольких частей:

– VPN демон PPTP;
– менеджер управления PPTP соединениями.

1.3 Ход работы

1.3.1 Установка необходимых пакетов для создания виртуальной частной сети

В состав необходимых пакетов входят:

– ppp (ftp://ftp.samba.org/pub/ppp/);
– pptpd (http://poptop.sourceforge.net/).

Данные пакеты уже могут быть установлены в системе, в таком случае данный этап работы является не обязательным. В противном случае и в случае необходимости обновит уже установленные версии пакетов их необходимо загрузить из сети и установить. Существует несколько способов установки пакетов в систему: при помощи менеджера пакетов используемого дистрибутива Linux (apt-get, yum и т.д.) (загрузка и установка будут происходить автоматически); загрузка и установка уже собранного пакета для используемого дистрибутива Linux (deb, rpm и т.д.); загрузка исходного кода пакета с последующей его сборкой и установкой.

Рассмотрим наиболее универсальный вариант – установка пакетов из исходных кодов. Для этого необходимо загрузить исходные коды пакетов, обычно помимо официального ресурса разработчика в сети существуют множество зеркал хранящих разные версии пакетов.

1.3.2 Загрузка пакетов

Пакеты необходимые для лабораторной работы можно загрузить с их официальных сайтов:

– ppp (ftp://ftp.samba.org/pub/ppp/);
– pptpd (http://poptop.sourceforge.net/).

1.3.3 Распаковка пакетов

Разархивирование можно сделать командами:

```
$ tar xvf ppp-<version>.tar.gz
$ tar xvf pptpd-<version>.tar.gz
```

В результате разархивирования должны быть созданы одноимённые с именами пакетов папки без префикса tar.gz.

1.3.4 Установка пакета ppp

Некоторые команды обычно необходимо выполнять с правами администратора, это отображено сепаратором # приглашения командной строки.

Войдите в корневую папку пакета ppp-<version>. Сконфигурируйте дальнейший процесс установки пакета ppp, это можно сделать следующими командами:

```
$ cd ppp-<version>
$ ./configure --prefix=/usr
```

Перед сборкой необходимо удостоверится в наличии gcc и make в системе, большинство дистрибутивов имеют одноименные пакеты. Выполним сборку ПО:

```
$ make
```

Выполним установку ПО:

```
# make install
# make install-etcppp
```

1.3.5 Установка пакета pptpd

Войдите в корневую папку пакета pptpd-<version>. Сконфигурируйте дальнейший процесс установки пакета pptpd, это можно сделать следующими командами:

```
cd pptp-<version>
$ ./configure --prefix=/ --exec-prefix=/usr --datarootdir=/usr/share
```

Забегая на перед следует отметить, что POPTOP скорее всего не заработает в паре с установленным ppp. Связано это с тем, что при сборке плагина POPTOP pptpd-logwtmp.so в нем указывается версия ppp, с которой необходимо работать и которая, обычно, не совпадает с версией ppp, который используется в системе. Для исправления этой ошибки необходимо перед сборкой отредактировать файл plugins/patchlevel.h, заменив в нем значение переменной VERSION на версию ppp:

```
/* upstream patchlevel.h,v 1.60 2004/01/13 04:46:52 paulus Exp */
/* $Id: patchlevel.h,v 1.4 2005/02/24 01:25:34 quozl Exp $ */

#define VERSION         "ppp-version-here"
#define DATE            "13 Jan 2004"
```

Выполним сборку ПО:

```
$ make
```

Выполним установку ПО:

```
# make install
```

1.3.6 Создание конфигурационных файлов pptp

Примеры конфигурационных файлов пакета pptpd находятся в папке pptpd-<version>/samples. Скопируем их в каталог с системными конфигурационными файлами:

```
# cp samples/chap-secrets samples/options.pptpd /etc/ppp
# cp samples/pptpd.conf /etc
```

Описание параметров конфигурационных и командных файлов изложены в соответствующих man страницах.

Пример конфигурационного файла /etc/pptpd.conf:

```
##############################################################################
# $Id: pptpd.conf,v 1.10 2006/09/04 23:30:57 quozl Exp $
#
# Sample Poptop configuration file /etc/pptpd.conf
#
# Changes are effective when pptpd is restarted.
##############################################################################
# TAG: ppp
#       Path to the pppd program, default '/usr/sbin/pppd' on Linux
#
#ppp /usr/sbin/pppd

# TAG: option
#       Specifies the location of the PPP options file.
#       By default PPP looks in '/etc/ppp/options'
#
option /etc/ppp/options.pptpd

# TAG: debug
#       Turns on (more) debugging to syslog
#
#debug

# TAG: stimeout
#       Specifies timeout (in seconds) on starting ctrl connection
#
# stimeout 10

# TAG: noipparam
#       Suppress the passing of the client's IP address to PPP, which is
```

```
#       done by default otherwise.
#
#noipparam

# TAG: logwtmp
#       Use wtmp(5) to record client connections and disconnections.
#
logwtmp

# TAG: bcrelay <if>
#       Turns on broadcast relay to clients from interface <if>
#
#bcrelay eth1

# TAG: delegate
#       Delegates the allocation of client IP addresses to pppd.
#
#       Without this option, which is the default, pptpd manages the list of
#       IP addresses for clients and passes the next free address to pppd.
#       With this option, pptpd does not pass an address, and so pppd may use
#       radius or chap-secrets to allocate an address.
#
#delegate

# TAG: connections
#       Limits the number of client connections that may be accepted.
#
#       If pptpd is allocating IP addresses (e.g. delegate is not
#       used) then the number of connections is also limited by the
#       remoteip option.  The default is 100.
#connections 100

# TAG: localip
# TAG: remoteip
#       Specifies the local and remote IP address ranges.
#
#       These options are ignored if delegate option is set.
#
#       Any addresses work as long as the local machine takes care of the
#       routing.  But if you want to use MS-Windows networking, you should
#       use IP addresses out of the LAN address space and use the proxyarp
#       option in the pppd options file, or run bcrelay.
#
#       You can specify single IP addresses seperated by commas or you can
#       specify ranges, or both. For example:
#
#               192.168.0.234,192.168.0.245-249,192.168.0.254
#
#       IMPORTANT RESTRICTIONS:
#
#       1. No spaces are permitted between commas or within addresses.
#
#       2. If you give more IP addresses than the value of connections,
#          it will start at the beginning of the list and go until it
#          gets connections IPs.  Others will be ignored.
#
#       3. No shortcuts in ranges! ie. 234-8 does not mean 234 to 238,
#          you must type 234-238 if you mean this.
#
#       4. If you give a single localIP, that's ok - all local IPs will
#          be set to the given one. You MUST still give at least one remote
#          IP for each simultaneous client.
#
# (Recommended)
#localip 192.168.0.1
#remoteip 192.168.0.234-238,192.168.0.245
# or
localip 192.168.105.50,192.168.105.59
remoteip 192.168.105.60,192.168.105.69
```

Большинство параметров имеют комментарии и в пояснениях не нуждаются. Параметры localip и remoteip указывают диапазон адресов локальных и удаленных точек VPN тоннеля соответственно. Локальная точка тоннеля представляется виртуальным интерфейсом ppp0, создаваемым на клиентском хосте при подключении к PPTP серверу. Удаленная точка —

соответствующим виртуальным ppp интерфейсом на хосте с PPTP сервером. Данная схема проиллюстрирована на рисунке 1.2 в данных методических указаний.

Пример конфигурационного файла /etc/ppp/options.pptpd:

```
################################################################################
# $Id: options.pptpd,v 1.11 2005/12/29 01:21:09 quozl Exp $
#
# Sample Poptop PPP options file /etc/ppp/options.pptpd
# Options used by PPP when a connection arrives from a client.
# This file is pointed to by /etc/pptpd.conf option keyword.
# Changes are effective on the next connection.  See "man pppd".
#
# You are expected to change this file to suit your system.  As
# packaged, it requires PPP 2.4.2 and the kernel MPPE module.
################################################################################


# Authentication

# Name of the local system for authentication purposes
# (must match the second field in /etc/ppp/chap-secrets entries)
name pptpd

# Strip the domain prefix from the username before authentication.
# (applies if you use pppd with chapms-strip-domain patch)
#chapms-strip-domain


# Encryption
# (There have been multiple versions of PPP with encryption support,
# choose with of the following sections you will use.)


# BSD licensed ppp-2.4.2 upstream with MPPE only, kernel module ppp_mppe.o
# {{{
refuse-pap
refuse-chap
refuse-mschap
# Require the peer to authenticate itself using MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] authentication.
require-mschap-v2
# Require MPPE 128-bit encryption
# (note that MPPE requires the use of MSCHAP-V2 during authentication)
require-mppe-128
# }}}


# OpenSSL licensed ppp-2.4.1 fork with MPPE only, kernel module mppe.o
# {{{
#-chap
#-chapms
# Require the peer to authenticate itself using MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] authentication.
#+chapms-v2
# Require MPPE encryption
# (note that MPPE requires the use of MSCHAP-V2 during authentication)
#mppe-40        # enable either 40-bit or 128-bit, not both
#mppe-128
#mppe-stateless
# }}}


# Network and Routing

# If pppd is acting as a server for Microsoft Windows clients, this
# option allows pppd to supply one or two DNS (Domain Name Server)
# addresses to the clients.  The first instance of this option
# specifies the primary DNS address; the second instance (if given)
# specifies the secondary DNS address.
#ms-dns 10.0.0.1
#ms-dns 10.0.0.2

# If pppd is acting as a server for Microsoft Windows or "Samba"
# clients, this option allows pppd to supply one or two WINS (Windows
# Internet Name Services) server addresses to the clients.  The first
# instance of this option specifies the primary WINS address; the
```

```
# second instance (if given) specifies the secondary WINS address.
#ms-wins 10.0.0.3
#ms-wins 10.0.0.4

# Add an entry to this system's ARP [Address Resolution Protocol]
# table with the IP address of the peer and the Ethernet address of this
# system.  This will have the effect of making the peer appear to other
# systems to be on the local ethernet.
# (you do not need this if your PPTP server is responsible for routing
# packets to the clients -- James Cameron)
proxyarp

# Normally pptpd passes the IP address to pppd, but if pptpd has been
# given the delegate option in pptpd.conf or the --delegate command line
# option, then pppd will use chap-secrets or radius to allocate the
# client IP address.  The default local IP address used at the server
# end is often the same as the address of the server.  To override this,
# specify the local IP address here.
# (you must not use this unless you have used the delegate option)
#10.8.0.100


# Logging

# Enable connection debugging facilities.
# (see your syslog configuration for where pppd sends to)
#debug

# Print out all the option values which have been set.
# (often requested by mailing list to verify options)
#dump


# Miscellaneous

# Create a UUCP-style lock file for the pseudo-tty to ensure exclusive
# access.
lock

# Disable BSD-Compress compression
nobsdcomp

# Disable Van Jacobson compression
# (needed on some networks with Windows 9x/ME/XP clients, see posting to
# poptop-server on 14th April 2005 by Pawel Pokrywka and followups,
# http://marc.theaimsgroup.com/?t=111343175400006&r=1&w=2 )
novj
novjccomp

# turn off logging to stderr, since this may be redirected to pptpd,
# which may trigger a loopback
nologfd

# put plugins here
# (putting them higher up may cause them to sent messages to the pty)
```

1.3.7 Добавление клиента виртуальной частной сети

Клиенты PPTP сервера POPTOP идентифицируются посредством механизмов пакета ppp. Существует несколько способов для управления клиентами. Наиболее простой из них это редактирование конфигурационного файла с логинами и паролями пользователей. Более сложный посредством идентификации через RADIUS сервер – задание последующей лабораторной работы.

Рассмотрим способ добавления пользователя через конфигурационный файл. Пример конфигурационного файла /etc/ppp/chap-secrets (для POPTOP важно указать в колонке «server» значение параметра «name» из файла /etc/ppp/options.pptp):

```
# Secrets for authentication using CHAP
```

```
# client        server  secret          IP addresses
linux           pptpd   test123         *
windows         pptpd   qwerty1         *
```

1.3.8 Тестирование работы сервера pptp

PPTP сервер РОРТОР можно запустить следующей командой:

```
# pptpd
```

Проверьте, запускается ли сервер вообще:

```
$ ps ax | grep pptp
 3245 ?        Ss     0:00 pptpd
```

По умолчания сервер прослушивает порт 1723, таким образом работоспособность сервера можно проверить следующей командой:

```
$ netstat -nl | grep 1723
tcp        0      0 0.0.0.0:1723            0.0.0.0:*               LISTEN
```

При установленных в конфигурационных файлах опция debug, работы сервера детально журналируется демоном syslog, по умолчания сообщения сервера будут помещаться в файл /etc/log/daemon.log. Динамически просматривать изменение данного файла можно запустив следующую команду:

```
# tail -d /etc/log/daemon.log
```

1.3.9 Настройка клиента созданной виртуальной частной сети из ОС Windows

Рассмотрим пример подключения ОС Windows 7. Для создания VPN подключения необходимо выполнить следующие шаги:

а) Пуск → Панель управления → Центр управления сетями и общим доступом → Настройка нового подключения к сети;
б) Будет запущен мастер создания нового подключения к сети;
в) Выбрать «Подключение к рабочему месту»;
г) Выбрать «Использовать мое подключение к Интернету»;
д) Выбрать «Отложить настройку подключения к Интернету»;
е) В поле «Интернет-адрес» ввести IP адрес сервера РОРТОР;
ж) Ввести имя пользователя и пароль;
з) Готово.

После запуска созданного подключения будет затребован логин и пароль пользователя, необходимо ввести данные введённые на 7м шаге.

1.3.10 Настройка клиента созданной виртуальной частной сети из ОС Linux

Для примера рассмотрим механизм подключения ОС Linux посредством ручного редактирования конфигурационных файлов и запуска демона pppd. Для подключения клиента должен быть установлен пакет pptp-

linux (pptpclient). В случае его отсутствия необходимо произвести его установку одним из описанных в пункте 1 методом.

Подключение может быть выполнено следующей командой (для отладки в конце можно дописать опции nodetach и debug):

```
# pppd call vpn
```

Где vpn имя файла с настройками подключения находящегося в папке /etc/ppp/peers/.

Пример конфигурационного файла vpn:

```
pty "pptp <poptop_server_ip_address> --nolaunchpppd"
name linux
file /etc/ppp/options.pptp
defaultroute
```

Пример конфигурационного файла /etc/ppp/options.pptp:

```
################################################################################
# $Id: options.pptp,v 1.3 2006/03/26 23:11:05 quozl Exp $
#
# Sample PPTP PPP options file /etc/ppp/options.pptp
# Options used by PPP when a connection is made by a PPTP client.
# This file can be referred to by an /etc/ppp/peers file for the tunnel.
# Changes are effective on the next connection.  See "man pppd".
#
# You are expected to change this file to suit your system.  As
# packaged, it requires PPP 2.4.2 or later from http://ppp.samba.org/
# and the kernel MPPE module available from the CVS repository also on
# http://ppp.samba.org/, which is packaged for DKMS as kernel_ppp_mppe.
################################################################################

# Lock the port
lock

# Authentication
# We don't need the tunnel server to authenticate itself
noauth

# We won't do PAP, EAP, CHAP, or MSCHAP, but we will accept MSCHAP-V2
# (you may need to remove these refusals if the server is not using MPPE)
refuse-pap
refuse-eap
refuse-chap
refuse-mschap

# Compression
# Turn off compression protocols we know won't be used
nobsdcomp
nodeflate

# Encryption
# (There have been multiple versions of PPP with encryption support,
# choose with of the following sections you will use.  Note that MPPE
# requires the use of MSCHAP-V2 during authentication)

# http://ppp.samba.org/ the PPP project version of PPP by Paul Mackarras
# ppp-2.4.2 or later with MPPE only, kernel module ppp_mppe.o
# {{{
# Require MPPE 128-bit encryption
require-mppe-128
# }}}

# http://polbox.com/h/hs001/ fork from PPP project by Jan Dubiec
# ppp-2.4.2 or later with MPPE and MPPC, kernel module ppp_mppe_mppc.o
# {{{
# Require MPPE 128-bit encryption
#mppe required,stateless
# }}}
```

Пример конфигурационного файла chap-secrets:

```
# Secrets for authentication using CHAP
# client        server  secret                  IP addresses
linux           *       test123                 *
```

1.3.11 Включение форвардинга пакетов

По умолчанию возможность форвардинга пакетов между интерфейсами в ОС Linux отключена. Для её активации необходимо отредактировать файл /etc/sysctl.conf параметров ядра, где параметру net.ipv4.ip_forward присвоить значение 1. Внесённое изменение вступит в силу после перезагрузки. Для немедленного применения новых параметров ядра необходимо выполнить команду:

```
# sysctl -p
```

1.3.12 Создание частной приватной сети

Сеть состоит из трех хостов:

– HOST1 настроенный сервер PPTP под ОС Linux;
– HOST2 настроенный vpn клиент под ОС Windows;
– HOST3 настроенный vpn клиент под ОС Linux.

Сетевой интерфейс HOST1 должен иметь IP адрес маршрутизируемый в сети университета и IP адрес (на том же интерфейсе – алиас) тестовой сети, не маршрутизируемой в сети университета.

Сетевой интерфейс HOST2, HOST3 должен иметь IP адрес тестовой сети. После установления VPN подключения HOST2, HOST3 должны получить IP адрес из сети университета, этим самым получив доступ к её ресурсам. Схема до подключения показана на рисунке 1.1, после подключения – на рисунке 1.2.

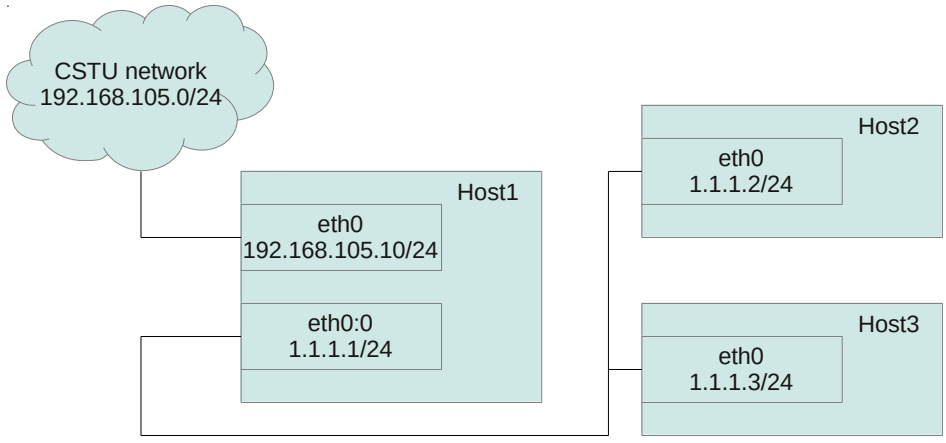

Рисунок 1.1 — Пример схемы сети до подключения HOST2, HOST3 к VPN серверу

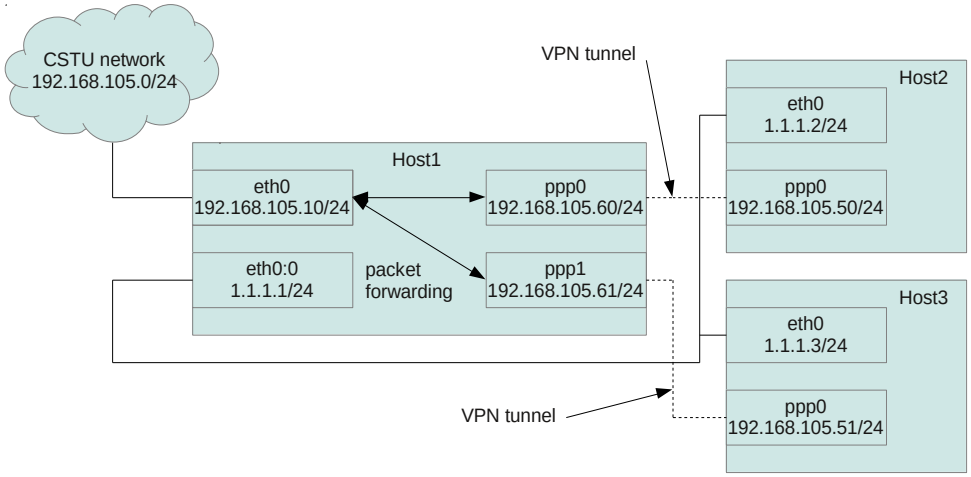

Рисунок 1.2 — Пример схемы сети после подключения HOST2, HOST3 к VPN серверу

1.4 Содержание отчета

Отчёт должен содержать ход выполнения 7го пункта с необходимыми конфигурационными файлами, комментариями по внесённым изменениям, листинга команды ifconfig до и после установления VPN соединения на сервера и на клиенте, скрипта отображающего корректную работу работу созданной сети.

1.5 Контрольные вопросы

1. Причины использования VPN сетей?
2. Назовите основные протоколы туннелирования.
3. Какие наиболее распространённые протоколы идентификации в PPP? В чем их разница?
4. Какие пакеты необходимо установить в систему для возможности создания PPTP сервера? Какие их основные чати?
5. В чём разница между PPP и PPTP?
6. Как добавить пользователя на сервере?
7. Как подключиться клиентом из ОС Windows/Linux?
8. Какие основные параметры конфигурационного файла PPTP?
9. Возможно ли использовать шифрование траффика в тоннеле? Какие протоколы для этого используются?
10. Возможно ли создание более одного VPN соединения? Если да, то как это сделать?

2 Самостоятельная работа №2. Настройка аутентификации, авторизации и учёта VPN пользователей посредством RADIUS сервера FreeRADIUS, с применение модуля rlm_sql

2.1 Цель работы

Установка и конфигурирование RADIUS сервера FreeRADIUS и сервера БД PostgreSQL под операционной системой Linux. Конфигурирование PPTP сервера POPTOP. Тестирование работы VPN клиентов под операционными системами Linux и Windows.

2.2 Теоретические сведения

Произведём определение терминов употребляемых в данной лабораторной работе:

Идентификация (лат. identifico – отождествлять): в философии – установление тождественности неизвестного объекта известному на основании совпадения признаков, опознание; в компьютерной безопасности – процесс сообщения субъектом своего имени или номера, с целью отличить данный субъект от других субъектов (например, одна из типичных систем идентификации – штрих-код); под идентификацией также иногда понимается аутентификация – процесс подтверждения подлинности пользователя; идентификация в психологии – вид психологической защиты.

Аутентификация (англ. Authentication) или идентификация, подтверждение подлинности – проверка соответствия субъекта и того, за кого он пытается себя выдать, с помощью некой уникальной информации (отпечатки пальцев, радужной оболочки глаз, голос и т. д.), в простейшем случае – с помощью имени (логина, от англ. login) и пароля. Процесс аутентификации основан на предположении о том, что пользователи имеют уникальные описывающие их данные и критерии доступа к ресурсам системы.

Авторизация (англ. Authorization) – процесс, а также результат процесса проверки необходимых параметров и предоставление определённых полномочий лицу или группе лиц на выполнение некоторых действий в различных системах с ограниченным доступом. Обычно процесс авторизации связан с процессом аутентификации (аутентифицированные пользователь авторизируется на разный тип доступа и возможные действия).

Учёт/Аккаунтинг (англ. Accounting) – процесс учета используемого сервиса. Данные учёта в дальнейшем обычно используются для получения статистики использования ресурса, начисления платы, построения прогнозов и так далее.

AAA – система позволяющая производить аутентификацию, авторизацию и учёт.

На данное время существует множество систем позволяющих производить аутентификацию, авторизацию и учёт. Они различаются: целевыми платформами, средствами реализации, возможностями масштабирования и наращивания, конкретными особенностями и так далее.

Центром всех этих систем аутентификации авторизации и учёта является не что иное, как протокол, посредством которого и решаются возлагаемые на эти системы задачи. В целом существующие протоколы можно разделить на две группы:

- те, которые реализуют все составляющие аутентификации авторизации и учёта (TACACS, XTACACS, TACACS+, RADIUS DIAMETER);
- те, которые реализуют часть этих составляющих, обычно только аутентификацию, или аутентификацию и авторизацию (ISAKMP/IKE (аутентификация), SASL (аутентификация), PolicyMarker (аутентификация и авторизация), KeyNote2 (аутентификация и авторизация)).

Для реализации систем AAA применяется первая группа протоколов.

Семейство протоколов TACACS (Terminal Access Controller Access Control System) в последнее время поддерживается компанией Cisco, онако в силу того что протокол контролируется конкретной организацией он не приобрёл популярности и применяется только в решениях которые строятся на базе оборудования этой компании.

Протокол RADIUS (Remote Authentication Dial In User Service) является клиент-серверным протоколом аутентификации, авторизации и учёта. Клиентом выступает NAS (Network Access Server) который производит запросы на аутентификацию своих клиентов. Сервер имеет доступ к данным с пользовательской информацией аутентификации. Данный протокол был разработан Livingston Enterprises приблизительно в 1989 и в дальнейшем поддерживался Мичигановским Университетом (Merit). В 1997 в IETF были внесены следующие RFC: Remote Access Dial In User Service (RADIUS) [RFC2138]; RADIUS Accounting [RFC2139]. Протокол поддерживается и используется многими производителями терминальных серверов, такими как Cisco, Ascend, Livingston и другими.

Протокол DIAMETER (DIAMETER = 2*RADIUS) предполагается использоваться для обеспечения политики безопасности, AAA и контроля ресурсов. Данный протокол позиционируется как замена более старому протоколу RADIUS, который до этого уже был широко распространён, с возможностью его полной эмуляции. Описание формата сообщений и транспорта используемого всеми расширениями или реализациями DIAMETER содержаться в RFC3588 и RFC4005. Расширение DIAMETER это расширение на базе описанного протокола для специальных сервисов или задач используемые DIAMETER как транспортного механизма.

Сводные характеристик данных протоколов приведены в таблицах 2.1-2.3.

Таблица 2.1 – Характеристики протокола TACACS+

Название	Значение
Транспортный протокол	TCP
Обмен сообщениями	запрос/ответ от клиента к серверу
Последовательная (hop-by-hop) безопасность	шифруется всё сообщение общим секретным ключом
Сквозная (end-to-end) безопасность	нет
Размер сообщения	заголовок (12 байт) + Nатрибута(1..N) * атрибут (8..255 байт)
Общее количество различных атрибутов	256
Возможность перенаправления запроса	нет

Таблица 2.2 – Характеристики протокола RADIUS

Название	Значение
Транспортный протокол	UDP
Обмен сообщениями	запрос/ответ от клиента к серверу
Последовательная (hop-by-hop) безопасность	шифруется только пароль общим секретным ключом
Сквозная (end-to-end) безопасность	нет
Размер сообщения	заголовок (12 байт) + Nатрибута(0..N) * атрибут (3..255 байт)
Общее количество различных атрибутов	256
Возможность перенаправления запроса	есть

Таблица 2.3 – Характеристики протокола DIAMETER

Название	Значение
Транспортный протокол	UDP или TCP
Обмен сообщениями	запрос/ответ от клиента к серверу, сообщения от сервера клиенту
Последовательная (hop-by-hop) безопасность	шифруется всё сообщение общим секретным ключом, не шифруется при использовании шифрования на IP уровне
Сквозная (end-to-end) безопасность	есть
Размер сообщения	заголовок (12 байт) + NAVP(0..N) * AVP (12..65400 байт)
Общее количество различных атрибутов	65535
Возможность перенаправления запроса	есть

Так как протокол RADIUS является базовым и, на данный момент, наиболее распространённым, то рассмотрим структуру его пакета и виды запросов и ответов протокола.

Данные между клиентом и сервером передаются посредством RADIUS пакета. Один RADIUS пакет инкапсулируется в UDP пакет. Каждый пакет содержит информацию, приведённую на рисунке 2.1.

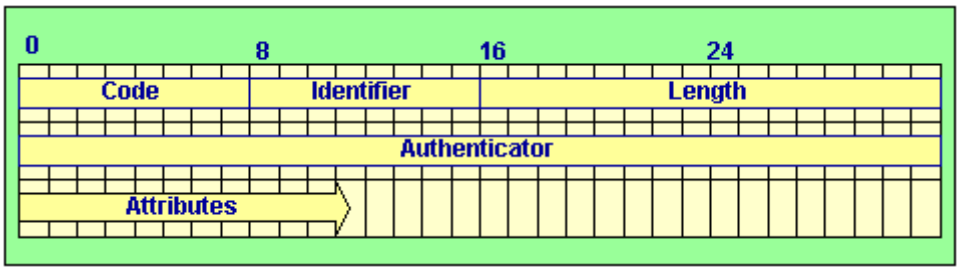

Рисунок 2.1 — Структура RADIUS пакета

Структура RADIUS пакета состоит из:

— Code – содержит код RADIUS команды/ответа;

— Indentifier – используется для сопоставления запроса и ответа не него;

— Length – длина пакет;

— Authenticator – используется для аутентификации ответа от RADIUS сервера, а также используется в алгоритме сокрытия пароля;

— Attributes – данные принадлежащие пакетам запроса и ответа, главной задачей атрибутов RADIUS является транспортировка информации между клиентами, серверами и прочими агентами RADIUS, некоторые атрибуты могут включатся больше чем один раз, это зависит от специфики атрибута.

Длина списка атрибутов определяется длиной RADIUS пакета.

Общий формат атрибута предоставлен на рисунке 2.2.

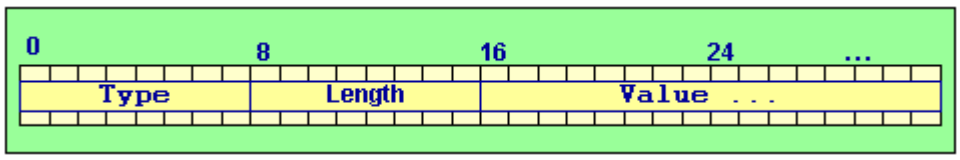

Рисунок 2.2 — Формат RADIUS аттрибута

Формат RADIUS аттрибута состоит из:

— Type – текущие значение данного поля приводятся в последних FRC, которые относятся к протоколу RADIUS. Значения 192-223 зарезервированы для экспериментального использования, значения 223-240 зарезервированы для реализационно-специфического использования, и значения 241-255 зарезервированы и не должны использоваться. Как RADIUS сервер так и RADIUS клиент могут игнорировать атрибуты неизвестного типа;

— Length – длина атрибута, включается поля Type, Length и Value;

— Value – значение атрибута, данное поле может быть нулевой или большей длины и содержит информацию специфичную для данного

атрибута, формат и длина данного поля определяется полями Type и Length.

Поле Value может быль следующих форматов:

— string – 0-253 октета (восьмиразрядного байта), причём не обязательно окончание строки 0ом (ASCII NUL) так как известна длина атрибута;
— address – 32х битное значение;
— integer – 32х битное значение;
— time – 32х битное значение, количество секунд от 00:00:00 GMT 1го января 1970го года, стандартные атрибуты не используют данный тип, он существует для предоставления возможности его использовать в атрибутах определяемых фирмой-поставщиком.

Стандартные типы атрибутов (User-Name, User-Password и так далее) и их форматы деятельно описаны в соответствующих RFC.

Взаимодействие посредством RADIUS использует парадигму запрос-ответ, запрос формируется клиентом и посылается серверу, ответ формируется сервером и посылается клиенту. Возможные следующие пары запрос-ответ:

— access-request, (client->server), запрос доступа пользователю к некоторому сервису. Возможные ответы на этот запрос:
— access-accept, (server->client), позитивный ответ на запрос access-request от клиента;
— access-reject, (server->client), негативный ответ на запрос access-request от клиента;
— access-challenge, (server->client), ответ на запрос access-reuquest, когда сервер ожидает некоторого дополнительного ответа от клиента на запрос access-request;
— accounting request, (client->server), запрос на сохранение учётной информации. Ответ на этот запрос:
— accounting response, (server->client), ответ клиенту в случае успешного сохранения учётной информации на сервере.

Типовой порядок доступа пользователя к сети посредством сетевого сервера доступа и дальнейшим отключением приведён на рисунке 2.3.

Для защиты сообщений клиент и сервер RADIUS обладают "общим секретом" (Shared Secret) или, проще говоря, ключом. При этом речь, как правило, идет о цепочке символов, имеющейся как на серверах, так и на клиентах RADIUS.

В данной лабораторной работе в качестве RADIUS сервера будем использовать GPL сервер FreeRADIUS, который является высоко производительным и хорошо конфигурируемым решением. Благодаря его стабильности на данный момент он применяется в многих разработках которые обеспечивают работу миллионов пользователей.

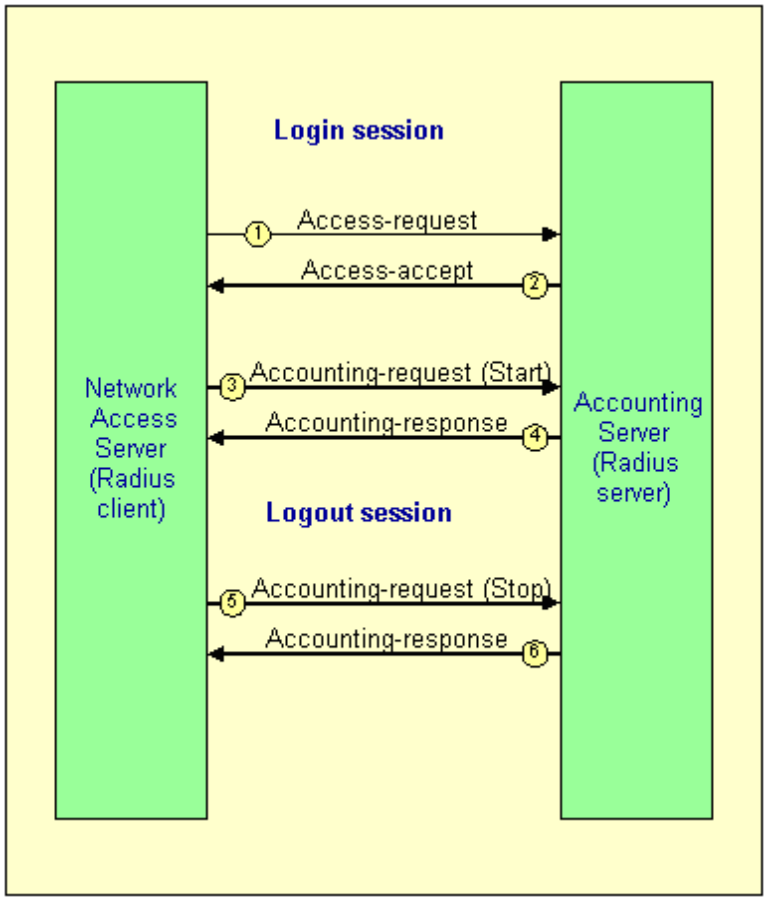

Рисунок 2.3 — Типовой порядок доступа пользователя к сети

Кратко рассмотрим рассмотрим как сервер FreeRADIUS обрабатывает запросы аутентификации, авторизации и учёта. Обычно запрос аутентификации полученный от NAS к FreeRADIUS обрабатывается в 2а этапа: авторизация и аутентификация. В случае сервера FreeRADIUS авторизации – это процесс получения информации о пользователе из внешних источников (файл, БД или LDAP), и проверка того что информации в запросе достаточно для авторизации пользователя. Авторизационные модули взаимодействуют с источниками данных, следовательно ldap, sql, файлы, passwd являются авторизационными модулями. Метод посредством которого будет происходить аутентификация выбирается на протяжении авторизационного этапа. Это позволяет строить гибкие политики аутентификации и авторизации, например запретить пользователю использовать некоторые методы аутентификации. Аутентификация – это процесс сравнения пользовательских параметров доступа в запросе с параметрами доступа хранимыми во внешних источнике данных сервера FreeRADIUS (файл, БД, LDAP и так далее).

На протяжении процесса авторизации и аутентификации FreeRADIUS поддерживает 3и списка RADIUS атрибутов: элементы запроса, конфигурационные элементы и элементы ответа. Атрибуты из аутентификационного RADIUS пакета включаются в список элементов запроса. Модули авторизации и аутентификации могут добавлять атрибуты в список элементов ответа, данных атрибуты будут добавлены в ответный

пакет RADIUS сервера NAS. Список конфигурационных элементов используется для внутренних операций сервера FreeRADIUS, например для передачи некоторых данных от модуля авторизации модулю аутентификации (пароль пользователя).

Как было сказано выше, сервер FreeRADIUS имеет модульную архитектуру, что предоставляет удобный механизм для добавления необходимой функциональности. Так например для работы с внешним источником данных в виде серверами БД используется соответствующий модуль (rlm_sql), который позволяет взаимодействовать с различными БД, такими например как db2, oracle, sybase, mysql, postgresql и др. В свою очередь взаимодействие RADIUS сервера с сервером БД позволяет удобно управлять данными авторизации, аутентификации и учёта.

В качестве сервера БД будем использовать сервер PostgreSQL, который является развитой объектно реляционной системой управления базами данных поддерживающей расширенное подмножества стандарта SQL, включая транзакции, внешние ключи, подзапросы, триггеры, пользовательские типы и функции.

2.3 Ход работы

2.3.1 Установка необходимых пакетов

В состав необходимых пакетов входят:

– postgresql (http://www.postgresql.org/download/);
– freeradius (http://freeradius.org/download.html);
– все пакеты установленные и настроенные в ходе первой лабораторной работы.

Данные пакеты уже могут быть установлены в системе, в таком случае данный этап работы является не обязательным. В противном случае и в случае необходимости обновит уже установленные версии пакетов их необходимо загрузить из сети и установить. Существует несколько способов установки пакетов в систему: при помощи менеджера пакетов используемого дистрибутива Linux (apt-get, yum и т.д.) (загрузка и установка будут происходить автоматически); загрузка и установка уже собранного пакета для используемого дистрибутива Linux (deb, rpm и т.д.); загрузка исходного кода пакета с последующей его сборкой и установкой.

Рассмотрим наиболее универсальный вариант – установка пакетов из исходных кодов. Для этого необходимо загрузить исходные коды пакетов, обычно помимо официального ресурса разработчика в сети существуют множество зеркал хранящих разные версии пакетов.

2.3.2 Загрузка пакетов

Пакеты необходимые для лабораторной работы можно загрузить с их официальных сайтов:

– postgresql (http://www.postgresql.org/ftp/source/);
– freeradius (http://freeradius.org/download.html).

2.3.3 Распаковка пакетов

Разархивирование можно сделать командами:

```
$ tar xvf postgresql-<version>.tar.gz
$ tar xvf freeradius-server-<version>.tar.gz
```

В результате разархивирования должны быть созданы одноимённые с именами пакетов папки без префикса tar.gz.

2.3.4 Установка пакета postgresql

Войдите в корневую папку пакета postgresql-<version>. *Необходимо наличие bison, flex — большинство дистрибутивов имеют одноименные пакеты, а также development версий библиотек libz, libreadline (zlib-devel, readline-devel для Fedora, RHEL; libz-dev, libreadline-dev для Debian, Ubuntu).* Для конфигурирования, компиляции и установки выполните следующие команды:

```
$ cd postgresql-<version>
$ ./configure --prefix=/ --exec-prefix=/usr --datarootdir=/usr/share
$ make
$ sudo -s
# make install
# useradd -d /dev/null -s /bin/false -M postgres
# mkdir -p /var/lib/pgsql/data
# chown -R postgres /var/lib/pgsql
# sudo -u postgres initdb -D /var/lib/pgsql/data/
# sudo -u postgres pg_ctl -D /var/lib/pgsql/data -l
/var/lib/pgsql/data/postgres.log start
```

Проверьте работоспособность сервера:

```
# ps ax | grep postgres
14041 pts/0    S      0:00 /usr/bin/postgres -D /var/lib/pgsql/data
14043 ?        Ss     0:00 postgres: writer process
14044 ?        Ss     0:00 postgres: wal writer process
14045 ?        Ss     0:00 postgres: autovacuum launcher process
14046 ?        Ss     0:00 postgres: stats collector process
# netstat -nl | grep 5432
tcp        0      0 127.0.0.1:5432          0.0.0.0:*            LISTEN
# sudo -u postgres createdb test
root@serpentarium:/home/student/postgresql-9.1.1# sudo -u postgres psql
test
psql (9.1.1)
Type "help" for help.

test=# \q
```

Более детальную инструкцию по установке postgresql из исходных кодов можно получить в файле postgresql-<version>/INSTALL.

2.3.5 Установка пакета freeradius

Войдите в корневую папку пакета freeradius-<version>. *Необходимо наличие development версий библиотек libdb, libpq (libdb-devel, libpqxx-devel для Fedora, RHEL; libdb-dev, libpqxx-dev для Debian, Ubuntu).* Соберите пакет freeradius, это можно сделать следующими командами:

```
$ cd freeradius-server-<version>
$ $ ./configure --prefix=/ --exec-prefix=/usr --datarootdir=/usr/share
$ make
$ sudo -s
# make install
# mkdir -p /var/run/radiusd
```

Более детальную инструкцию по установке freeradius из исходных кодов можно получить в файле freeradius-<version>/INSTALL.

2.3.6 Создание БД для модуля rlm_sql/rlm_sql_postresql сервера FreeRADIUS

Создание пользователя БД PostgreSQL можно сделать командами createuser и createdb. Команды могут быть выполнены только пользователем БД имеющем права на создание пользователей и создание БД. По умолчанию таким пользователем является postgres:

```
$ sudo -u postgres createuser -E -P freeradius
Enter password for new role: freeradius
Enter it again: freeradius
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
$ sudo -u postgres createdb -O freeradius freeradius
```

Так как в схеме БД используемой модулем rlm_sql/rlm_sql_postgresql применяются хранимые процедуры то в созданной БД необходимо объявить язык plpgsql, это можно сделать следующей командой (обычно создан по умолчанию):

```
$ sudo -u postgres createlang plpgsql freeradius
```

По умолчанию проинициализированный репозиторий БД настроен так что аутентификация всех пользователей и БД производиться методом ident sameuser (соответствие пользователей PostgreSQL, владельца БД и пользователя ОС системы). Поэтому в конфигурационный файл методов аутентификации клиентов PosgtreSQL (по умолчанию /var/lib/pgsql/data/pg_hba.conf), необходимо перед всеми остальными настройками, добавить следующую строчку:

```
# TYPE  DATABASE        USER            ADDRESS             METHOD
host    freeradius      freeradius      127.0.0.1/32        password
```

И перезапустить сервер БД PostgreSQL:

```
# sudo -u postgres pg_ctl -D /var/lib/pgsql/data -l
/var/lib/pgsql/data/postgres.log restart
```

Завершающим этапом создание БД в сервере БД PostgreSQL для модуля rlm_sql/rlm_sql_postresql сервера FreeRADIUS, является создание таблиц, хранимых процедур, триггеров и т.д. Модуль rlm_sql позволяет самостоятельно придумать и создать схему БД, заполнить соответствующие поля запросов в конфигурационном файле модуля, и использовать модуль. Однако данный модуль также предоставляет стандартные схемы БД и

конфигурационные файлы для наиболее общераспространённых серверов БД, таких как oraclesql, posgtresql, mysql. Схемы БД (файлы db_oracle.sql, db_postgresql.sql, db_mysql.sql), просле установки, расположены в каталоге /etc/raddb/sql. Конфигурационные файлы (файлы oraclesql.conf, postgresql.conf) в директории /etc/raddb/.

Инициализацию созданной БД стандартной схемой БД для модуля rlm_sql/rlm_sql_postgresql можно сделать следующей командой:

```
$ sudo cp /etc/raddb/sql/postgresql/schema.sql /tmp
$ sudo chmod 777 /tmp/schema.sql
$ sudo -u postgres psql -h 127.0.0.1 -d freeradius -U freeradius -f
/tmp/schema.sql
```

2.3.7 Добавление пользователей в созданную БД

Стандартная схема БД для модуля rlm_sql/rlm_sql_postgresql отображает структуру конфигурационного файла сервера FreeRADIUS "users". Таким образом для того что бы понять какие данные необходимо добавить в БД, можно изучить man 5 users, и примеры приводимые в файле users.

Для добавления простейшего типа пользователя, где имя пользователя сопоставляется с его паролем, без проверки типа RADIUS пакета и без добавления дополнительных атрибутов ответного пакета, можно выполнить следующий SQL оператор:

```
$ sudo -u postgres psql -h 127.0.0.1 freeradius freeradius
freeradius=> insert into radcheck (UserName, Attribute, op, Value)
values ('linux', 'User-Password', '==', 'test123');
INSERT 0 1
freeradius=> insert into radcheck (UserName, Attribute, op, Value)
values ('windows', 'User-Password', '==', 'qwerty1');
INSERT 0 1
freeradius=> \q
```

2.3.8 Конфигурирование сервера FreeRADIUS

При установки пакета FreeRADIUS также устанавливаются его конфигурационные файлы по умолчанию (обычно это директория /etc/raddb/), если в устанавливаемой директории конфигурационные файлы уже присутствуют (при обновлении пакета) они не перетираются, при этом выводиться предупреждение о файлах которые не были установлены.

Основным конфигурационным файлом сервера FreeRADIUS является /etc/raddb/radiusd.conf, указать другой путь к данному конфигурационному файлу можно через параметры запуска radiusd. Описание структуры и параметров данного конфигурационного и модулей сервера FreeRADIUS можно прочитать в соответствующих man страницах (man 5 radiusd.conf и т.д.).

Так как основной конфигурационный файл имеет довольно большие размеры в силу хорошего описания всех его элементов, то приведём его, обозначив только те места где в него вносились изменения.

Пример конфигурационного файла /etc/raddb/radiusd.conf:

```
# -*- text -*-
##
## radiusd.conf -- FreeRADIUS server configuration file.
##
##      http://www.freeradius.org/
##      $Id$
##

####################################################################
#
#       Read "man radiusd" before editing this file.  See the section
#       titled DEBUGGING.  It outlines a method where you can quickly
#       obtain the configuration you want, without running into
#       trouble.
#
#       Run the server in debugging mode, and READ the output.
#
#               $ radiusd -X
#
#       We cannot emphasize this point strongly enough.  The vast
#       majority of problems can be solved by carefully reading the
#       debugging output, which includes warnings about common issues,
#       and suggestions for how they may be fixed.
#
#       There may be a lot of output, but look carefully for words like:
#       "warning", "error", "reject", or "failure".  The messages there
#       will usually be enough to guide you to a solution.
#
#       If you are going to ask a question on the mailing list, then
#       explain what you are trying to do, and include the output from
#       debugging mode (radiusd -X).  Failure to do so means that all
#       of the responses to your question will be people telling you
#       to "post the output of radiusd -X".

####################################################################
#
#       The location of other config files and logfiles are declared
#       in this file.
#
#       Also general configuration for modules can be done in this
#       file, it is exported through the API to modules that ask for
#       it.
#
#       See "man radiusd.conf" for documentation on the format of this
#       file.  Note that the individual configuration items are NOT
#       documented in that "man" page.  They are only documented here,
#       in the comments.
#
#       As of 2.0.0, FreeRADIUS supports a simple processing language
#       in the "authorize", "authenticate", "accounting", etc. sections.
#       See "man unlang" for details.
#

prefix = /
exec_prefix = /usr
sysconfdir = ${prefix}/etc
localstatedir = ${prefix}/var
sbindir = ${exec_prefix}/sbin
logdir = ${localstatedir}/log/radius
raddbdir = ${sysconfdir}/raddb
radacctdir = ${logdir}/radacct

#
#  name of the running server.  See also the "-n" command-line option.
name = radiusd

#  Location of config and logfiles.
confdir = ${raddbdir}
run_dir = ${localstatedir}/run/${name}

# Should likely be ${localstatedir}/lib/radiusd
db_dir = ${raddbdir}

#
# libdir: Where to find the rlm_* modules.
#
#   This should be automatically set at configuration time.
```

```
#
#   If the server builds and installs, but fails at execution time
#   with an 'undefined symbol' error, then you can use the libdir
#   directive to work around the problem.
#
#   The cause is usually that a library has been installed on your
#   system in a place where the dynamic linker CANNOT find it.  When
#   executing as root (or another user), your personal environment MAY
#   be set up to allow the dynamic linker to find the library.  When
#   executing as a daemon, FreeRADIUS MAY NOT have the same
#   personalized configuration.
#
#   To work around the problem, find out which library contains that symbol,
#   and add the directory containing that library to the end of 'libdir',
#   with a colon separating the directory names.  NO spaces are allowed.
#
#   e.g. libdir = /usr/local/lib:/opt/package/lib
#
#   You can also try setting the LD_LIBRARY_PATH environment variable
#   in a script which starts the server.
#
#   If that does not work, then you can re-configure and re-build the
#   server to NOT use shared libraries, via:
#
#       ./configure --disable-shared
#       make
#       make install
#
libdir = ${exec_prefix}/lib

#  pidfile: Where to place the PID of the RADIUS server.
#
#  The server may be signalled while it's running by using this
#  file.
#
#  This file is written when ONLY running in daemon mode.
#
#  e.g.:  kill -HUP `cat /var/run/radiusd/radiusd.pid`
#
pidfile = ${run_dir}/${name}.pid

#  chroot: directory where the server does "chroot".
#
#  The chroot is done very early in the process of starting the server.
#  After the chroot has been performed it switches to the "user" listed
#  below (which MUST be specified).  If "group" is specified, it switchs
#  to that group, too.  Any other groups listed for the specified "user"
#  in "/etc/group" are also added as part of this process.
#
#  The current working directory (chdir / cd) is left *outside* of the
#  chroot until all of the modules have been initialized.  This allows
#  the "raddb" directory to be left outside of the chroot.  Once the
#  modules have been initialized, it does a "chdir" to ${logdir}.  This
#  means that it should be impossible to break out of the chroot.
#
#  If you are worried about security issues related to this use of chdir,
#  then simply ensure that the "raddb" directory is inside of the chroot,
#  end be sure to do "cd raddb" BEFORE starting the server.
#
#  If the server is statically linked, then the only files that have
#  to exist in the chroot are ${run_dir} and ${logdir}.  If you do the
#  "cd raddb" as discussed above, then the "raddb" directory has to be
#  inside of the chroot directory, too.
#
#chroot = /path/to/chroot/directory

# user/group: The name (or #number) of the user/group to run radiusd as.
#
#   If these are commented out, the server will run as the user/group
#   that started it.  In order to change to a different user/group, you
#   MUST be root ( or have root privleges ) to start the server.
#
#   We STRONGLY recommend that you run the server with as few permissions
#   as possible.  That is, if you're not using shadow passwords, the
#   user and group items below should be set to radius'.
#
#  NOTE that some kernels refuse to setgid(group) when the value of
#  (unsigned)group is above 60000; don't use group nobody on these systems!
#
#  On systems with shadow passwords, you might have to set 'group = shadow'
```

```
#  for the server to be able to read the shadow password file.  If you can
#  authenticate users while in debug mode, but not in daemon mode, it may be
#  that the debugging mode server is running as a user that can read the
#  shadow info, and the user listed below can not.
#
#  The server will also try to use "initgroups" to read /etc/groups.
#  It will join all groups where "user" is a member.  This can allow
#  for some finer-grained access controls.
#
#user = radius
#group = radius

#  max_request_time: The maximum time (in seconds) to handle a request.
#
#  Requests which take more time than this to process may be killed, and
#  a REJECT message is returned.
#
#  WARNING: If you notice that requests take a long time to be handled,
#  then this MAY INDICATE a bug in the server, in one of the modules
#  used to handle a request, OR in your local configuration.
#
#  This problem is most often seen when using an SQL database.  If it takes
#  more than a second or two to receive an answer from the SQL database,
#  then it probably means that you haven't indexed the database.  See your
#  SQL server documentation for more information.
#
#  Useful range of values: 5 to 120
#
max_request_time = 30

#  cleanup_delay: The time to wait (in seconds) before cleaning up
#  a reply which was sent to the NAS.
#
#  The RADIUS request is normally cached internally for a short period
#  of time, after the reply is sent to the NAS.  The reply packet may be
#  lost in the network, and the NAS will not see it.  The NAS will then
#  re-send the request, and the server will respond quickly with the
#  cached reply.
#
#  If this value is set too low, then duplicate requests from the NAS
#  MAY NOT be detected, and will instead be handled as seperate requests.
#
#  If this value is set too high, then the server will cache too many
#  requests, and some new requests may get blocked.  (See 'max_requests'.)
#
#  Useful range of values: 2 to 10
#
cleanup_delay = 5

#  max_requests: The maximum number of requests which the server keeps
#  track of.  This should be 256 multiplied by the number of clients.
#  e.g. With 4 clients, this number should be 1024.
#
#  If this number is too low, then when the server becomes busy,
#  it will not respond to any new requests, until the 'cleanup_delay'
#  time has passed, and it has removed the old requests.
#
#  If this number is set too high, then the server will use a bit more
#  memory for no real benefit.
#
#  If you aren't sure what it should be set to, it's better to set it
#  too high than too low.  Setting it to 1000 per client is probably
#  the highest it should be.
#
#  Useful range of values: 256 to infinity
#
max_requests = 1024

#  listen: Make the server listen on a particular IP address, and send
#  replies out from that address. This directive is most useful for
#  hosts with multiple IP addresses on one interface.
#
#  If you want the server to listen on additional addresses, or on
#  additionnal ports, you can use multiple "listen" sections.
#
#  Each section make the server listen for only one type of packet,
#  therefore authentication and accounting have to be configured in
#  different sections.
#
#  The server ignore all "listen" section if you are using '-i' and '-p'
```

```
#  on the command line.
#
listen {
        #  Type of packets to listen for.
        #  Allowed values are:
        #       auth    listen for authentication packets
        #       acct    listen for accounting packets
        #       proxy   IP to use for sending proxied packets
        #       detail  Read from the detail file.  For examples, see
        #               raddb/sites-available/copy-acct-to-home-server
        #       status  listen for Status-Server packets.  For examples,
        #               see raddb/sites-available/status
        #       coa     listen for CoA-Request and Disconnect-Request
        #               packets.  For examples, see the file
        #               raddb/sites-available/coa-server
        #
        type = auth

        #  Note: "type = proxy" lets you control the source IP used for
        #        proxying packets, with some limitations:
        #
        #    * A proxy listener CANNOT be used in a virtual server section.
        #    * You should probably set "port = 0".
        #    * Any "clients" configuration will be ignored.
        #
        #  See also proxy.conf, and the "src_ipaddr" configuration entry
        #  in the sample "home_server" section.  When you specify the
        #  source IP address for packets sent to a home server, the
        #  proxy listeners are automatically created.

        #  IP address on which to listen.
        #  Allowed values are:
        #       dotted quad (1.2.3.4)
        #       hostname    (radius.example.com)
        #       wildcard    (*)
        ipaddr = *

        #  OR, you can use an IPv6 address, but not both
        #  at the same time.
#       ipv6addr = ::   # any.  ::1 == localhost

        #  Port on which to listen.
        #  Allowed values are:
        #       integer port number (1812)
        #       0 means "use /etc/services for the proper port"
        port = 0

        #  Some systems support binding to an interface, in addition
        #  to the IP address.  This feature isn't strictly necessary,
        #  but for sites with many IP addresses on one interface,
        #  it's useful to say "listen on all addresses for eth0".
        #
        #  If your system does not support this feature, you will
        #  get an error if you try to use it.
        #
#       interface = eth0

        #  Per-socket lists of clients.  This is a very useful feature.
        #
        #  The name here is a reference to a section elsewhere in
        #  radiusd.conf, or clients.conf.  Having the name as
        #  a reference allows multiple sockets to use the same
        #  set of clients.
        #
        #  If this configuration is used, then the global list of clients
        #  is IGNORED for this "listen" section.  Take care configuring
        #  this feature, to ensure you don't accidentally disable a
        #  client you need.
        #
        #  See clients.conf for the configuration of "per_socket_clients".
        #
#       clients = per_socket_clients
}

#  This second "listen" section is for listening on the accounting
#  port, too.
#
listen {
        ipaddr = *
#       ipv6addr = ::
```

```
        port = 0
        type = acct
#       interface = eth0
#       clients = per_socket_clients
}

#  hostname_lookups: Log the names of clients or just their IP addresses
#  e.g., www.freeradius.org (on) or 206.47.27.232 (off).
#
#  The default is 'off' because it would be overall better for the net
#  if people had to knowingly turn this feature on, since enabling it
#  means that each client request will result in AT LEAST one lookup
#  request to the nameserver.   Enabling hostname_lookups will also
#  mean that your server may stop randomly for 30 seconds from time
#  to time, if the DNS requests take too long.
#
#  Turning hostname lookups off also means that the server won't block
#  for 30 seconds, if it sees an IP address which has no name associated
#  with it.
#
#  allowed values: {no, yes}
#
hostname_lookups = no

#  Core dumps are a bad thing.  This should only be set to 'yes'
#  if you're debugging a problem with the server.
#
#  allowed values: {no, yes}
#
allow_core_dumps = no

#  Regular expressions
#
#  These items are set at configure time.  If they're set to "yes",
#  then setting them to "no" turns off regular expression support.
#
#  If they're set to "no" at configure time, then setting them to "yes"
#  WILL NOT WORK.  It will give you an error.
#
regular_expressions     = yes
extended_expressions    = yes

#
#  Logging section.  The various "log_*" configuration items
#  will eventually be moved here.
#
log {
        #
        #  Destination for log messages.  This can be one of:
        #
        #       files - log to "file", as defined below.
        #       syslog - to syslog (see also the "syslog_facility", below.
        #       stdout - standard output
        #       stderr - standard error.
        #
        #  The command-line option "-X" over-rides this option, and forces
        #  logging to go to stdout.
        #
        destination = files

        #
        #  The logging messages for the server are appended to the
        #  tail of this file if destination == "files"
        #
        #  If the server is running in debugging mode, this file is
        #  NOT used.
        #
        file = ${logdir}/radius.log

        #
        #  If this configuration parameter is set, then log messages for
        #  a *request* go to this file, rather than to radius.log.
        #
        #  i.e. This is a log file per request, once the server has accepted
        #  the request as being from a valid client.  Messages that are
        #  not associated with a request still go to radius.log.
        #
        #  Not all log messages in the server core have been updated to use
        #  this new internal API.  As a result, some messages will still
        #  go to radius.log.  Please submit patches to fix this behavior.
```

```
        #
        #  The file name is expanded dynamically.  You should ONLY user
        #  server-side attributes for the filename (e.g. things you control).
        #  Using this feature MAY also slow down the server substantially,
        #  especially if you do thinks like SQL calls as part of the
        #  expansion of the filename.
        #
        #  The name of the log file should use attributes that don't change
        #  over the lifetime of a request, such as User-Name,
        #  Virtual-Server or Packet-Src-IP-Address.  Otherwise, the log
        #  messages will be distributed over multiple files.
        #
        #  Logging can be enabled for an individual request by a special
        #  dynamic expansion macro:  %{debug: 1}, where the debug level
        #  for this request is set to '1' (or 2, 3, etc.).  e.g.
        #
        #       ...
        #       update control {
        #              Tmp-String-0 = "%{debug:1}"
        #       }
        #       ...
        #
        #  The attribute that the value is assigned to is unimportant,
        #  and should be a "throw-away" attribute with no side effects.
        #
        #requests = ${logdir}/radiusd-%{%{Virtual-Server}:-DEFAULT}-%Y%m%d.log

        #
        #  Which syslog facility to use, if ${destination} == "syslog"
        #
        #  The exact values permitted here are OS-dependent.  You probably
        #  don't want to change this.
        #
        syslog_facility = daemon

        #  Log the full User-Name attribute, as it was found in the request.
        #
        # allowed values: {no, yes}
        #
        stripped_names = no

        #  Log authentication requests to the log file.
        #
        #  allowed values: {no, yes}
        #
        auth = no

        #  Log passwords with the authentication requests.
        #  auth_badpass  - logs password if it's rejected
        #  auth_goodpass - logs password if it's correct
        #
        #  allowed values: {no, yes}
        #
        auth_badpass = no
        auth_goodpass = no

        #  Log additional text at the end of the "Login OK" messages.
        #  for these to work, the "auth" and "auth_goopass" or "auth_badpass"
        #  configurations above have to be set to "yes".
        #
        #  The strings below are dynamically expanded, which means that
        #  you can put anything you want in them.  However, note that
        #  this expansion can be slow, and can negatively impact server
        #  performance.
        #
#       msg_goodpass = ""
#       msg_badpass = ""
}

#  The program to execute to do concurrency checks.
checkrad = ${sbindir}/checkrad

# SECURITY CONFIGURATION
#
#  There may be multiple methods of attacking on the server.  This
#  section holds the configuration items which minimize the impact
#  of those attacks
#
security {
        #
```

```
        #  max_attributes: The maximum number of attributes
        #  permitted in a RADIUS packet.  Packets which have MORE
        #  than this number of attributes in them will be dropped.
        #
        #  If this number is set too low, then no RADIUS packets
        #  will be accepted.
        #
        #  If this number is set too high, then an attacker may be
        #  able to send a small number of packets which will cause
        #  the server to use all available memory on the machine.
        #
        #  Setting this number to 0 means "allow any number of attributes"
        max_attributes = 200

        #
        #  reject_delay: When sending an Access-Reject, it can be
        #  delayed for a few seconds.  This may help slow down a DoS
        #  attack.  It also helps to slow down people trying to brute-force
        #  crack a users password.
        #
        #  Setting this number to 0 means "send rejects immediately"
        #
        #  If this number is set higher than 'cleanup_delay', then the
        #  rejects will be sent at 'cleanup_delay' time, when the request
        #  is deleted from the internal cache of requests.
        #
        #  Useful ranges: 1 to 5
        reject_delay = 1

        #
        #  status_server: Whether or not the server will respond
        #  to Status-Server requests.
        #
        #  When sent a Status-Server message, the server responds with
        #  an Access-Accept or Accounting-Response packet.
        #
        #  This is mainly useful for administrators who want to "ping"
        #  the server, without adding test users, or creating fake
        #  accounting packets.
        #
        #  It's also useful when a NAS marks a RADIUS server "dead".
        #  The NAS can periodically "ping" the server with a Status-Server
        #  packet.  If the server responds, it must be alive, and the
        #  NAS can start using it for real requests.
        #
        #  See also raddb/sites-available/status
        #
        status_server = yes
}

# PROXY CONFIGURATION
#
#  proxy_requests: Turns proxying of RADIUS requests on or off.
#
#  The server has proxying turned on by default.  If your system is NOT
#  set up to proxy requests to another server, then you can turn proxying
#  off here.  This will save a small amount of resources on the server.
#
#  If you have proxying turned off, and your configuration files say
#  to proxy a request, then an error message will be logged.
#
#  To disable proxying, change the "yes" to "no", and comment the
#  $INCLUDE line.
#
#  allowed values: {no, yes}
#
proxy_requests  = yes
$INCLUDE proxy.conf


# CLIENTS CONFIGURATION
#
#  Client configuration is defined in "clients.conf".
#

#  The 'clients.conf' file contains all of the information from the old
#  'clients' and 'naslist' configuration files.  We recommend that you
#  do NOT use 'client's or 'naslist', although they are still
#  supported.
#
```

```
#  Anything listed in 'clients.conf' will take precedence over the
#  information from the old-style configuration files.
#
$INCLUDE clients.conf


# THREAD POOL CONFIGURATION
#
#  The thread pool is a long-lived group of threads which
#  take turns (round-robin) handling any incoming requests.
#
#  You probably want to have a few spare threads around,
#  so that high-load situations can be handled immediately.  If you
#  don't have any spare threads, then the request handling will
#  be delayed while a new thread is created, and added to the pool.
#
#  You probably don't want too many spare threads around,
#  otherwise they'll be sitting there taking up resources, and
#  not doing anything productive.
#
#  The numbers given below should be adequate for most situations.
#
thread pool {
        #  Number of servers to start initially --- should be a reasonable
        #  ballpark figure.
        start_servers = 5

        #  Limit on the total number of servers running.
        #
        #  If this limit is ever reached, clients will be LOCKED OUT, so it
        #  should NOT BE SET TOO LOW.  It is intended mainly as a brake to
        #  keep a runaway server from taking the system with it as it spirals
        #  down...
        #
        #  You may find that the server is regularly reaching the
        #  'max_servers' number of threads, and that increasing
        #  'max_servers' doesn't seem to make much difference.
        #
        #  If this is the case, then the problem is MOST LIKELY that
        #  your back-end databases are taking too long to respond, and
        #  are preventing the server from responding in a timely manner.
        #
        #  The solution is NOT do keep increasing the 'max_servers'
        #  value, but instead to fix the underlying cause of the
        #  problem: slow database, or 'hostname_lookups=yes'.
        #
        #  For more information, see 'max_request_time', above.
        #
        max_servers = 32

        #  Server-pool size regulation.  Rather than making you guess
        #  how many servers you need, FreeRADIUS dynamically adapts to
        #  the load it sees, that is, it tries to maintain enough
        #  servers to handle the current load, plus a few spare
        #  servers to handle transient load spikes.
        #
        #  It does this by periodically checking how many servers are
        #  waiting for a request.  If there are fewer than
        #  min_spare_servers, it creates a new spare.  If there are
        #  more than max_spare_servers, some of the spares die off.
        #  The default values are probably OK for most sites.
        #
        min_spare_servers = 3
        max_spare_servers = 10

        #  When the server receives a packet, it places it onto an
        #  internal queue, where the worker threads (configured above)
        #  pick it up for processing.  The maximum size of that queue
        #  is given here.
        #
        #  When the queue is full, any new packets will be silently
        #  discarded.
        #
        #  The most common cause of the queue being full is that the
        #  server is dependent on a slow database, and it has received
        #  a large "spike" of traffic.  When that happens, there is
        #  very little you can do other than make sure the server
        #  receives less traffic, or make sure that the database can
        #  handle the load.
        #
```

```
#       max_queue_size = 65536

        #  There may be memory leaks or resource allocation problems with
        #  the server.  If so, set this value to 300 or so, so that the
        #  resources will be cleaned up periodically.
        #
        #  This should only be necessary if there are serious bugs in the
        #  server which have not yet been fixed.
        #
        #  '0' is a special value meaning 'infinity', or 'the servers never
        #  exit'
        max_requests_per_server = 0
}

# MODULE CONFIGURATION
#
#  The names and configuration of each module is located in this section.
#
#  After the modules are defined here, they may be referred to by name,
#  in other sections of this configuration file.
#
modules {
        #
        #  Each module has a configuration as follows:
        #
        #       name [ instance ] {
        #               config_item = value
        #               ...
        #       }
        #
        #  The 'name' is used to load the 'rlm_name' library
        #  which implements the functionality of the module.
        #
        #  The 'instance' is optional.  To have two different instances
        #  of a module, it first must be referred to by 'name'.
        #  The different copies of the module are then created by
        #  inventing two 'instance' names, e.g. 'instance1' and 'instance2'
        #
        #  The instance names can then be used in later configuration
        #  INSTEAD of the original 'name'.  See the 'radutmp' configuration
        #  for an example.
        #

        #
        #  As of 2.0.5, most of the module configurations are in a
        #  sub-directory.  Files matching the regex /[a-zA-Z0-9_.]+/
        #  are loaded.  The modules are initialized ONLY if they are
        #  referenced in a processing section, such as authorize,
        #  authenticate, accounting, pre/post-proxy, etc.
        #
        $INCLUDE ${confdir}/modules/

        #  Extensible Authentication Protocol
        #
        #  For all EAP related authentications.
        #  Now in another file, because it is very large.
        #
        $INCLUDE eap.conf

        #  Include another file that has the SQL-related configuration.
        #  This is another file only because it tends to be big.
        #
        $INCLUDE sql.conf

        #
        #  This module is an SQL enabled version of the counter module.
        #
        #  Rather than maintaining seperate (GDBM) databases of
        #  accounting info for each counter, this module uses the data
        #  stored in the raddacct table by the sql modules. This
        #  module NEVER does any database INSERTs or UPDATEs.  It is
        #  totally dependent on the SQL module to process Accounting
        #  packets.
        #
#       $INCLUDE sql/mysql/counter.conf

        #
        #  IP addresses managed in an SQL table.
        #
#       $INCLUDE sqlippool.conf
```

```
}

# Instantiation
#
#  This section orders the loading of the modules.  Modules
#  listed here will get loaded BEFORE the later sections like
#  authorize, authenticate, etc. get examined.
#
#  This section is not strictly needed.  When a section like
#  authorize refers to a module, it's automatically loaded and
#  initialized.  However, some modules may not be listed in any
#  of the following sections, so they can be listed here.
#
#  Also, listing modules here ensures that you have control over
#  the order in which they are initalized.  If one module needs
#  something defined by another module, you can list them in order
#  here, and ensure that the configuration will be OK.
#
instantiate {
        #
        #  Allows the execution of external scripts.
        #  The entire command line (and output) must fit into 253 bytes.
        #
        #  e.g. Framed-Pool = `%{exec:/bin/echo foo}`
        exec

        #
        #  The expression module doesn't do authorization,
        #  authentication, or accounting.  It only does dynamic
        #  translation, of the form:
        #
        #       Session-Timeout = `%{expr:2 + 3}`
        #
        #  So the module needs to be instantiated, but CANNOT be
        #  listed in any other section.  See 'doc/rlm_expr' for
        #  more information.
        #
        expr

        #
        # We add the counter module here so that it registers
        # the check-name attribute before any module which sets
        # it
#       daily
        expiration
        logintime

        # subsections here can be thought of as "virtual" modules.
        #
        # e.g. If you have two redundant SQL servers, and you want to
        # use them in the authorize and accounting sections, you could
        # place a "redundant" block in each section, containing the
        # exact same text.  Or, you could uncomment the following
        # lines, and list "redundant_sql" in the authorize and
        # accounting sections.
        #
        #redundant redundant_sql {
        #       sql1
        #       sql2
        #}
}

######################################################################
#
#       Policies that can be applied in multiple places are listed
#       globally.  That way, they can be defined once, and referred
#       to multiple times.
#
######################################################################
$INCLUDE policy.conf

######################################################################
#
#       Load virtual servers.
#
#       This next $INCLUDE line loads files in the directory that
#       match the regular expression: /[a-zA-Z0-9_.]+/
#
#       It allows you to define new virtual servers simply by placing
#       a file into the raddb/sites-enabled/ directory.
```

```
#
$INCLUDE sites-enabled/

######################################################################
#
#       All of the other configuration sections like "authorize {}",
#       "authenticate {}", "accounting {}", have been moved to the
#       the file:
#
#               raddb/sites-available/default
#
#       This is the "default" virtual server that has the same
#       configuration as in version 1.0.x and 1.1.x.  The default
#       installation enables this virtual server.  You should
#       edit it to create policies for your local site.
#
#       For more documentation on virtual servers, see:
#
#               raddb/sites-available/README
#
######################################################################
```

В конфигурационном файле настроек БД, необходимо ввести данные относительно созданной БД, пользователя сервера БД, его пароля и адреса сервера БД.

Пример конфигурационного файла sql.conf:

```
# -*- text -*-
##
## sql.conf -- SQL modules
##
##      $Id$

######################################################################
#
#  Configuration for the SQL module
#
#  The database schemas and queries are located in subdirectories:
#
#       sql/DB/schema.sql       Schema
#       sql/DB/dialup.conf      Basic dialup (including policy) queries
#       sql/DB/counter.conf     counter
#       sql/DB/ippool.conf      IP Pools in SQL
#       sql/DB/ippool.sql       schema for IP pools.
#
#  Where "DB" is mysql, mssql, oracle, or postgresql.
#

sql {
        #
        #  Set the database to one of:
        #
        #       mysql, mssql, oracle, postgresql
        #
        database = "postgresql"

        #
        #  Which FreeRADIUS driver to use.
        #
        driver = "rlm_sql_${database}"

        # Connection info:
        server = "127.0.0.1"
        port = 5432
        login = "freeradius"
        password = "freeradius"

        # Database table configuration for everything except Oracle
        radius_db = "freeradius"
        # If you are using Oracle then use this instead
        # radius_db =
"(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=localhost)(PORT=1521))(CONNECT_DATA=(SID=your_sid)))"

        # If you want both stop and start records logged to the
        # same SQL table, leave this as is.  If you want them in
        # different tables, put the start table in acct_table1
        # and stop table in acct_table2
```

```
        acct_table1 = "radacct"
        acct_table2 = "radacct"

        # Allow for storing data after authentication
        postauth_table = "radpostauth"

        authcheck_table = "radcheck"
        authreply_table = "radreply"

        groupcheck_table = "radgroupcheck"
        groupreply_table = "radgroupreply"

        # Table to keep group info
        usergroup_table = "radusergroup"

        # If set to 'yes' (default) we read the group tables
        # If set to 'no' the user MUST have Fall-Through = Yes in the radreply table
        # read_groups = yes

        # Remove stale session if checkrad does not see a double login
        deletestalesessions = yes

        # Print all SQL statements when in debug mode (-x)
        sqltrace = no
        sqltracefile = ${logdir}/sqltrace.sql

        # number of sql connections to make to server
        num_sql_socks = 5

        # number of seconds to dely retrying on a failed database
        # connection (per_socket)
        connect_failure_retry_delay = 60

        # lifetime of an SQL socket.  If you are having network issues
        # such as TCP sessions expiring, you may need to set the socket
        # lifetime.  If set to non-zero, any open connections will be
        # closed "lifetime" seconds after they were first opened.
        lifetime = 0

        # Maximum number of queries used by an SQL socket.  If you are
        # having issues with SQL sockets lasting "too long", you can
        # limit the number of queries performed over one socket.  After
        # "max_qeuries", the socket will be closed.  Use 0 for "no limit".
        max_queries = 0

        # Set to 'yes' to read radius clients from the database ('nas' table)
        # Clients will ONLY be read on server startup.  For performance
        # and security reasons, finding clients via SQL queries CANNOT
        # be done "live" while the server is running.
        #
        #readclients = yes

        # Table to keep radius client info
        nas_table = "nas"

        # Read driver-specific configuration
        $INCLUDE sql/${database}/dialup.conf
}
```

"Общие секреты" (ключи) сервера FreeRADIUS и NAS серверов с которыми он взаимодействует находятся в конфигурационном файле clients.conf. Как и все конфигурационные файлы FreeRADIUS, данный файл хорошо прокомментирован. Для выполнения лабораторной работы достаточно его настроек по умолчанию.

Пример конфигурационного файла /etc/raddb/clients.conf:

```
# -*- text -*-
##
## clients.conf -- client configuration directives
##
##      $Id$

#######################################################################
#
#  Define RADIUS clients (usually a NAS, Access Point, etc.).
```

```
#
#  Defines a RADIUS client.
#
#  '127.0.0.1' is another name for 'localhost'.  It is enabled by default,
#  to allow testing of the server after an initial installation.  If you
#  are not going to be permitting RADIUS queries from localhost, we suggest
#  that you delete, or comment out, this entry.
#
#


#
#  Each client has a "short name" that is used to distinguish it from
#  other clients.
#
#  In version 1.x, the string after the word "client" was the IP
#  address of the client.  In 2.0, the IP address is configured via
#  the "ipaddr" or "ipv6addr" fields.  For compatibility, the 1.x
#  format is still accepted.
#
client localhost {
        #  Allowed values are:
        #       dotted quad (1.2.3.4)
        #       hostname    (radius.example.com)
        ipaddr = 127.0.0.1

        #  OR, you can use an IPv6 address, but not both
        #  at the same time.
#       ipv6addr = ::   # any.  ::1 == localhost

        #
        #  A note on DNS:  We STRONGLY recommend using IP addresses
        #  rather than host names.  Using host names means that the
        #  server will do DNS lookups when it starts, making it
        #  dependent on DNS.  i.e. If anything goes wrong with DNS,
        #  the server won't start!
        #
        #  The server also looks up the IP address from DNS once, and
        #  only once, when it starts.  If the DNS record is later
        #  updated, the server WILL NOT see that update.
        #

        #  One client definition can be applied to an entire network.
        #  e.g. 127/8 should be defined with "ipaddr = 127.0.0.0" and
        #  "netmask = 8"
        #
        #  If not specified, the default netmask is 32 (i.e. /32)
        #
        #  We do NOT recommend using anything other than 32.  There
        #  are usually other, better ways to achieve the same goal.
        #  Using netmasks of other than 32 can cause security issues.
        #
        #  You can specify overlapping networks (127/8 and 127.0/16)
        #  In that case, the smallest possible network will be used
        #  as the "best match" for the client.
        #
        #  Clients can also be defined dynamically at run time, based
        #  on any criteria.  e.g. SQL lookups, keying off of NAS-Identifier,
        #  etc.
        #  See raddb/sites-available/dynamic-clients for details.
        #

#       netmask = 32

        #
        #  The shared secret use to "encrypt" and "sign" packets between
        #  the NAS and FreeRADIUS.  You MUST change this secret from the
        #  default, otherwise it's not a secret any more!
        #
        #  The secret can be any string, up to 8k characters in length.
        #
        #  Control codes can be entered vi octal encoding,
        #       e.g. "\101\102" == "AB"
        #  Quotation marks can be entered by escaping them,
        #       e.g. "foo\"bar"
        #
        #  A note on security:  The security of the RADIUS protocol
        #  depends COMPLETELY on this secret!  We recommend using a
        #  shared secret that is composed of:
        #
```

```
        #       upper case letters
        #       lower case letters
        #       numbers
        #
        #  And is at LEAST 8 characters long, preferably 16 characters in
        #  length.  The secret MUST be random, and should not be words,
        #  phrase, or anything else that is recognizable.
        #
        #  The default secret below is only for testing, and should
        #  not be used in any real environment.
        #
        secret          = testing123

        #
        #  Old-style clients do not send a Message-Authenticator
        #  in an Access-Request.  RFC 5080 suggests that all clients
        #  SHOULD include it in an Access-Request.  The configuration
        #  item below allows the server to require it.  If a client
        #  is required to include a Message-Authenticator and it does
        #  not, then the packet will be silently discarded.
        #
        #  allowed values: yes, no
        require_message_authenticator = no

        #
        #  The short name is used as an alias for the fully qualified
        #  domain name, or the IP address.
        #
        #  It is accepted for compatibility with 1.x, but it is no
        #  longer necessary in 2.0
        #
#       shortname       = localhost

        #
        # the following three fields are optional, but may be used by
        # checkrad.pl for simultaneous use checks
        #

        #
        # The nastype tells 'checkrad.pl' which NAS-specific method to
        #  use to query the NAS for simultaneous use.
        #
        #  Permitted NAS types are:
        #
        #       cisco
        #       computone
        #       livingston
        #       max40xx
        #       multitech
        #       netserver
        #       pathras
        #       patton
        #       portslave
        #       tc
        #       usrhiper
        #       other           # for all other types

        #
        nastype     = other     # localhost isn't usually a NAS...

        #
        #  The following two configurations are for future use.
        #  The 'naspasswd' file is currently used to store the NAS
        #  login name and password, which is used by checkrad.pl
        #  when querying the NAS for simultaneous use.
        #
#       login       = !root
#       password    = someadminpas

        #
        #  As of 2.0, clients can also be tied to a virtual server.
        #  This is done by setting the "virtual_server" configuration
        #  item, as in the example below.
        #
#       virtual_server = home1

        #
        #  A pointer to the "home_server_pool" OR a "home_server"
        #  section that contains the CoA configuration for this
        #  client.  For an example of a coa home server or pool,
```

```
        #  see raddb/sites-available/originate-coa
#       coa_server = coa
}

# IPv6 Client
#client ::1 {
#       secret          = testing123
#       shortname       = localhost
#}
#
# All IPv6 Site-local clients
#client fe80::/16 {
#       secret          = testing123
#       shortname       = localhost
#}

#client some.host.org {
#       secret          = testing123
#       shortname       = localhost
#}

#
#  You can now specify one secret for a network of clients.
#  When a client request comes in, the BEST match is chosen.
#  i.e. The entry from the smallest possible network.
#
#client 192.168.0.0/24 {
#       secret          = testing123-1
#       shortname       = private-network-1
#}
#
#client 192.168.0.0/16 {
#       secret          = testing123-2
#       shortname       = private-network-2
#}


#client 10.10.10.10 {
#       # secret and password are mapped through the "secrets" file.
#       secret      = testing123
#       shortname   = liv1
#       # the following three fields are optional, but may be used by
#       # checkrad.pl for simultaneous usage checks
#       nastype     = livingston
#       login       = !root
#       password    = someadminpas
#}

#############################################################################
#
#  Per-socket client lists.  The configuration entries are exactly
#  the same as above, but they are nested inside of a section.
#
#  You can have as many per-socket client lists as you have "listen"
#  sections, or you can re-use a list among multiple "listen" sections.
#
#  Un-comment this section, and edit a "listen" section to add:
#  "clients = per_socket_clients".  That IP address/port combination
#  will then accept ONLY the clients listed in this section.
#
#clients per_socket_clients {
#       client 192.168.3.4 {
#               secret = testing123
#        }
#}
```

Чтобы FreeRADIUS обращался к БД для получения логина и пароля авторизируемого пользователя, а также для внесения записей аккаунтинга, необходимо раскомментировать строки «sql» в файле sites-enabled/default в секциях authorize и authenticate соответственно. SQL запросы к БД хранятся в файле /etc/raddb/sql/postgresql/dialup.conf.

Пример конфигурационного файла /etc/raddb/sites-enabled/default:

```
######################################################################
#
#       As of 2.0.0, FreeRADIUS supports virtual hosts using the
#       "server" section, and configuration directives.
#
#       Virtual hosts should be put into the "sites-available"
#       directory.  Soft links should be created in the "sites-enabled"
#       directory to these files.  This is done in a normal installation.
#
#       If you are using 802.1X (EAP) authentication, please see also
#       the "inner-tunnel" virtual server.  You wll likely have to edit
#       that, too, for authentication to work.
#
#       $Id$
#
######################################################################
#
#       Read "man radiusd" before editing this file.  See the section
#       titled DEBUGGING.  It outlines a method where you can quickly
#       obtain the configuration you want, without running into
#       trouble.  See also "man unlang", which documents the format
#       of this file.
#
#       This configuration is designed to work in the widest possible
#       set of circumstances, with the widest possible number of
#       authentication methods.  This means that in general, you should
#       need to make very few changes to this file.
#
#       The best way to configure the server for your local system
#       is to CAREFULLY edit this file.  Most attempts to make large
#       edits to this file will BREAK THE SERVER.  Any edits should
#       be small, and tested by running the server with "radiusd -X".
#       Once the edits have been verified to work, save a copy of these
#       configuration files somewhere.  (e.g. as a "tar" file).  Then,
#       make more edits, and test, as above.
#
#       There are many "commented out" references to modules such
#       as ldap, sql, etc.  These references serve as place-holders.
#       If you need the functionality of that module, then configure
#       it in radiusd.conf, and un-comment the references to it in
#       this file.  In most cases, those small changes will result
#       in the server being able to connect to the DB, and to
#       authenticate users.
#
######################################################################

#
#       In 1.x, the "authorize", etc. sections were global in
#       radiusd.conf.  As of 2.0, they SHOULD be in a server section.
#
#       The server section with no virtual server name is the "default"
#       section.  It is used when no server name is specified.
#
#       We don't indent the rest of this file, because doing so
#       would make it harder to read.
#

#  Authorization. First preprocess (hints and huntgroups files),
#  then realms, and finally look in the "users" file.
#
#  Any changes made here should also be made to the "inner-tunnel"
#  virtual server.
#
#  The order of the realm modules will determine the order that
#  we try to find a matching realm.
#
#  Make *sure* that 'preprocess' comes before any realm if you
#  need to setup hints for the remote radius server
authorize {
        #
        #  Security settings.  Take a User-Name, and do some simple
        #  checks on it, for spaces and other invalid characters.  If
        #  it looks like the user is trying to play games, reject it.
        #
        #  This should probably be enabled by default.
        #
        #  See policy.conf for the definition of the filter_username policy.
        #
#       filter_username
```

```
        #
        #  The preprocess module takes care of sanitizing some bizarre
        #  attributes in the request, and turning them into attributes
        #  which are more standard.
        #
        #  It takes care of processing the 'raddb/hints' and the
        #  'raddb/huntgroups' files.
        preprocess

        #
        #  If you want to have a log of authentication requests,
        #  un-comment the following line, and the 'detail auth_log'
        #  section, above.
#       auth_log

        #
        #  The chap module will set 'Auth-Type := CHAP' if we are
        #  handling a CHAP request and Auth-Type has not already been set
        chap

        #
        #  If the users are logging in with an MS-CHAP-Challenge
        #  attribute for authentication, the mschap module will find
        #  the MS-CHAP-Challenge attribute, and add 'Auth-Type := MS-CHAP'
        #  to the request, which will cause the server to then use
        #  the mschap module for authentication.
        mschap

        #
        #  If you have a Cisco SIP server authenticating against
        #  FreeRADIUS, uncomment the following line, and the 'digest'
        #  line in the 'authenticate' section.
        digest

        #
        #  The WiMAX specification says that the Calling-Station-Id
        #  is 6 octets of the MAC.  This definition conflicts with
        #  RFC 3580, and all common RADIUS practices.  Un-commenting
        #  the "wimax" module here means that it will fix the
        #  Calling-Station-Id attribute to the normal format as
        #  specified in RFC 3580 Section 3.21
#       wimax

        #
        #  Look for IPASS style 'realm/', and if not found, look for
        #  '@realm', and decide whether or not to proxy, based on
        #  that.
#       IPASS

        #
        #  If you are using multiple kinds of realms, you probably
        #  want to set "ignore_null = yes" for all of them.
        #  Otherwise, when the first style of realm doesn't match,
        #  the other styles won't be checked.
        #
        suffix
#       ntdomain

        #
        #  This module takes care of EAP-MD5, EAP-TLS, and EAP-LEAP
        #  authentication.
        #
        #  It also sets the EAP-Type attribute in the request
        #  attribute list to the EAP type from the packet.
        #
        #  As of 2.0, the EAP module returns "ok" in the authorize stage
        #  for TTLS and PEAP.  In 1.x, it never returned "ok" here, so
        #  this change is compatible with older configurations.
        #
        #  The example below uses module failover to avoid querying all
        #  of the following modules if the EAP module returns "ok".
        #  Therefore, your LDAP and/or SQL servers will not be queried
        #  for the many packets that go back and forth to set up TTLS
        #  or PEAP.  The load on those servers will therefore be reduced.
        #
        eap {
                ok = return
        }

        #
```

```
        #  Pull crypt'd passwords from /etc/passwd or /etc/shadow,
        #  using the system API's to get the password.  If you want
        #  to read /etc/passwd or /etc/shadow directly, see the
        #  passwd module in radiusd.conf.
        #
#       unix

        #
        #  Read the 'users' file
        files

        #
        #  Look in an SQL database.  The schema of the database
        #  is meant to mirror the "users" file.
        #
        #  See "Authorization Queries" in sql.conf
        sql

        #
        #  If you are using /etc/smbpasswd, and are also doing
        #  mschap authentication, the un-comment this line, and
        #  configure the 'etc_smbpasswd' module, above.
#       etc_smbpasswd

        #
        #  The ldap module will set Auth-Type to LDAP if it has not
        #  already been set
#       ldap

        #
        #  Enforce daily limits on time spent logged in.
#       daily

        #
        # Use the checkval module
#       checkval

        expiration
        logintime

        #
        #  If no other module has claimed responsibility for
        #  authentication, then try to use PAP.  This allows the
        #  other modules listed above to add a "known good" password
        #  to the request, and to do nothing else.  The PAP module
        #  will then see that password, and use it to do PAP
        #  authentication.
        #
        #  This module should be listed last, so that the other modules
        #  get a chance to set Auth-Type for themselves.
        #
        pap

        #
        #  If "status_server = yes", then Status-Server messages are passed
        #  through the following section, and ONLY the following section.
        #  This permits you to do DB queries, for example.  If the modules
        #  listed here return "fail", then NO response is sent.
        #
#       Autz-Type Status-Server {
#
#       }
}


#  Authentication.
#
#
#  This section lists which modules are available for authentication.
#  Note that it does NOT mean 'try each module in order'.  It means
#  that a module from the 'authorize' section adds a configuration
#  attribute 'Auth-Type := FOO'.  That authentication type is then
#  used to pick the apropriate module from the list below.
#

#  In general, you SHOULD NOT set the Auth-Type attribute.  The server
#  will figure it out on its own, and will do the right thing.  The
#  most common side effect of erroneously setting the Auth-Type
#  attribute is that one authentication method will work, but the
#  others will not.
```

```
#
#  The common reasons to set the Auth-Type attribute by hand
#  is to either forcibly reject the user (Auth-Type := Reject),
#  or to or forcibly accept the user (Auth-Type := Accept).
#
#  Note that Auth-Type := Accept will NOT work with EAP.
#
#  Please do not put "unlang" configurations into the "authenticate"
#  section.  Put them in the "post-auth" section instead.  That's what
#  the post-auth section is for.
#
authenticate {
        #
        #  PAP authentication, when a back-end database listed
        #  in the 'authorize' section supplies a password.  The
        #  password can be clear-text, or encrypted.
        Auth-Type PAP {
                pap
        }

        #
        #  Most people want CHAP authentication
        #  A back-end database listed in the 'authorize' section
        #  MUST supply a CLEAR TEXT password.  Encrypted passwords
        #  won't work.
        Auth-Type CHAP {
                chap
        }

        #
        #  MSCHAP authentication.
        Auth-Type MS-CHAP {
                mschap
        }

        #
        #  If you have a Cisco SIP server authenticating against
        #  FreeRADIUS, uncomment the following line, and the 'digest'
        #  line in the 'authorize' section.
        digest

        #
        #  Pluggable Authentication Modules.
#       pam

        #
        #  See 'man getpwent' for information on how the 'unix'
        #  module checks the users password.  Note that packets
        #  containing CHAP-Password attributes CANNOT be authenticated
        #  against /etc/passwd!  See the FAQ for details.
        #
        #  For normal "crypt" authentication, the "pap" module should
        #  be used instead of the "unix" module.  The "unix" module should
        #  be used for authentication ONLY for compatibility with legacy
        #  FreeRADIUS configurations.
        #
        unix

        # Uncomment it if you want to use ldap for authentication
        #
        # Note that this means "check plain-text password against
        # the ldap database", which means that EAP won't work,
        # as it does not supply a plain-text password.
#       Auth-Type LDAP {
#               ldap
#       }

        #
        #  Allow EAP authentication.
        eap

        #
        #  The older configurations sent a number of attributes in
        #  Access-Challenge packets, which wasn't strictly correct.
        #  If you want to filter out these attributes, uncomment
        #  the following lines.
        #
#       Auth-Type eap {
#               eap {
#                       handled = 1
```

```
        #               }
        #               if (handled && (Response-Packet-Type == Access-Challenge)) {
        #                       attr_filter.access_challenge.post-auth
        #                       handled  # override the "updated" code from attr_filter
        #               }
        #       }
        }


        #
        #  Pre-accounting.  Decide which accounting type to use.
        #
        preacct {
                preprocess

                #
                #  Session start times are *implied* in RADIUS.
                #  The NAS never sends a "start time".  Instead, it sends
                #  a start packet, *possibly* with an Acct-Delay-Time.
                #  The server is supposed to conclude that the start time
                #  was "Acct-Delay-Time" seconds in the past.
                #
                #  The code below creates an explicit start time, which can
                #  then be used in other modules.
                #
                #  The start time is: NOW - delay - session_length
                #

        #         update request {
        #               FreeRADIUS-Acct-Session-Start-Time = "%{expr: %l - %{%{Acct-Session-
Time}:-0} - %{%{Acct-Delay-Time}:-0}}"
        #       }


                #
                #  Ensure that we have a semi-unique identifier for every
                #  request, and many NAS boxes are broken.
                acct_unique

                #
                #  Look for IPASS-style 'realm/', and if not found, look for
                #  '@realm', and decide whether or not to proxy, based on
                #  that.
                #
                #  Accounting requests are generally proxied to the same
                #  home server as authentication requests.
        #       IPASS
                suffix
        #       ntdomain

                #
                #  Read the 'acct_users' file
                files
        }

        #
        #  Accounting.  Log the accounting data.
        #
        accounting {
                #
                #  Create a 'detail'ed log of the packets.
                #  Note that accounting requests which are proxied
                #  are also logged in the detail file.
                detail
        #       daily

                #  Update the wtmp file
                #
                #  If you don't use "radlast", you can delete this line.
                unix

                #
                #  For Simultaneous-Use tracking.
                #
                #  Due to packet losses in the network, the data here
                #  may be incorrect.  There is little we can do about it.
                radutmp
        #       sradutmp

                #  Return an address to the IP Pool when we see a stop record.
```

```
#       main_pool

        #
        #  Log traffic to an SQL database.
        #
        #  See "Accounting queries" in sql.conf
        sql

        #
        #  If you receive stop packets with zero session length,
        #  they will NOT be logged in the database.  The SQL module
        #  will print a message (only in debugging mode), and will
        #  return "noop".
        #
        #  You can ignore these packets by uncommenting the following
        #  three lines.  Otherwise, the server will not respond to the
        #  accounting request, and the NAS will retransmit.
        #
#       if (noop) {
#               ok
#       }

        #
        #  Instead of sending the query to the SQL server,
        #  write it into a log file.
        #
#       sql_log

        #  Cisco VoIP specific bulk accounting
#       pgsql-voip

        # For Exec-Program and Exec-Program-Wait
        exec

        #  Filter attributes from the accounting response.
        attr_filter.accounting_response

        #
        #  See "Autz-Type Status-Server" for how this works.
        #
#       Acct-Type Status-Server {
#
#       }
}


#  Session database, used for checking Simultaneous-Use. Either the radutmp
#  or rlm_sql module can handle this.
#  The rlm_sql module is *much* faster
session {
        radutmp

        #
        #  See "Simultaneous Use Checking Queries" in sql.conf
#       sql
}


#  Post-Authentication
#  Once we KNOW that the user has been authenticated, there are
#  additional steps we can take.
post-auth {
        #  Get an address from the IP Pool.
#       main_pool

        #
        #  If you want to have a log of authentication replies,
        #  un-comment the following line, and the 'detail reply_log'
        #  section, above.
#       reply_log

        #
        #  After authenticating the user, do another SQL query.
        #
        #  See "Authentication Logging Queries" in sql.conf
#       sql

        #
        #  Instead of sending the query to the SQL server,
        #  write it into a log file.
```

```
        #
#       sql_log

        #
        #  Un-comment the following if you have set
        #  'edir_account_policy_check = yes' in the ldap module sub-section of
        #  the 'modules' section.
        #
#       ldap

        # For Exec-Program and Exec-Program-Wait
        exec

        #
        #  Calculate the various WiMAX keys.  In order for this to work,
        #  you will need to define the WiMAX NAI, usually via
        #
        #       update request {
        #              WiMAX-MN-NAI = "%{User-Name}"
        #       }
        #
        #  If you want various keys to be calculated, you will need to
        #  update the reply with "template" values.  The module will see
        #  this, and replace the template values with the correct ones
        #  taken from the cryptographic calculations.  e.g.
        #
        #       update reply {
        #               WiMAX-FA-RK-Key = 0x00
        #               WiMAX-MSK = "%{EAP-MSK}"
        #       }
        #
        #  You may want to delete the MS-MPPE-*-Keys from the reply,
        #  as some WiMAX clients behave badly when those attributes
        #  are included.  See "raddb/modules/wimax", configuration
        #  entry "delete_mppe_keys" for more information.
        #
#       wimax

        #  If there is a client certificate (EAP-TLS, sometimes PEAP
        #  and TTLS), then some attributes are filled out after the
        #  certificate verification has been performed.  These fields
        #  MAY be available during the authentication, or they may be
        #  available only in the "post-auth" section.
        #
        #  The first set of attributes contains information about the
        #  issuing certificate which is being used.  The second
        #  contains information about the client certificate (if
        #  available).
#
#       update reply {
#              Reply-Message += "%{TLS-Cert-Serial}"
#              Reply-Message += "%{TLS-Cert-Expiration}"
#              Reply-Message += "%{TLS-Cert-Subject}"
#              Reply-Message += "%{TLS-Cert-Issuer}"
#              Reply-Message += "%{TLS-Cert-Common-Name}"
#
#              Reply-Message += "%{TLS-Client-Cert-Serial}"
#              Reply-Message += "%{TLS-Client-Cert-Expiration}"
#              Reply-Message += "%{TLS-Client-Cert-Subject}"
#              Reply-Message += "%{TLS-Client-Cert-Issuer}"
#              Reply-Message += "%{TLS-Client-Cert-Common-Name}"
#       }


        #  If the WiMAX module did it's work, you may want to do more
        #  things here, like delete the MS-MPPE-*-Key attributes.
        #
        #       if (updated) {
        #               update reply {
        #                       MS-MPPE-Recv-Key !* 0x00
        #                       MS-MPPE-Send-Key !* 0x00
        #               }
        #       }

        #
        #  Access-Reject packets are sent through the REJECT sub-section of the
        #  post-auth section.
        #
        #  Add the ldap module name (or instance) if you have set
        #  'edir_account_policy_check = yes' in the ldap module configuration
```

```
        #
        Post-Auth-Type REJECT {
                # log failed authentications in SQL, too.
#               sql
                attr_filter.access_reject
        }
}

#
#  When the server decides to proxy a request to a home server,
#  the proxied request is first passed through the pre-proxy
#  stage.  This stage can re-write the request, or decide to
#  cancel the proxy.
#
#  Only a few modules currently have this method.
#
pre-proxy {
#       attr_rewrite

        #  Uncomment the following line if you want to change attributes
        #  as defined in the preproxy_users file.
#       files

        #  Uncomment the following line if you want to filter requests
        #  sent to remote servers based on the rules defined in the
        #  'attrs.pre-proxy' file.
#       attr_filter.pre-proxy

        #  If you want to have a log of packets proxied to a home
        #  server, un-comment the following line, and the
        #  'detail pre_proxy_log' section, above.
#       pre_proxy_log
}

#
#  When the server receives a reply to a request it proxied
#  to a home server, the request may be massaged here, in the
#  post-proxy stage.
#
post-proxy {

        #  If you want to have a log of replies from a home server,
        #  un-comment the following line, and the 'detail post_proxy_log'
        #  section, above.
#       post_proxy_log

#       attr_rewrite

        #  Uncomment the following line if you want to filter replies from
        #  remote proxies based on the rules defined in the 'attrs' file.
#       attr_filter.post-proxy

        #
        #  If you are proxying LEAP, you MUST configure the EAP
        #  module, and you MUST list it here, in the post-proxy
        #  stage.
        #
        #  You MUST also use the 'nostrip' option in the 'realm'
        #  configuration.  Otherwise, the User-Name attribute
        #  in the proxied request will not match the user name
        #  hidden inside of the EAP packet, and the end server will
        #  reject the EAP request.
        #
        eap

        #
        #  If the server tries to proxy a request and fails, then the
        #  request is processed through the modules in this section.
        #
        #  The main use of this section is to permit robust proxying
        #  of accounting packets.  The server can be configured to
        #  proxy accounting packets as part of normal processing.
        #  Then, if the home server goes down, accounting packets can
        #  be logged to a local "detail" file, for processing with
        #  radrelay.  When the home server comes back up, radrelay
        #  will read the detail file, and send the packets to the
        #  home server.
        #
        #  With this configuration, the server always responds to
        #  Accounting-Requests from the NAS, but only writes
```

```
        #  accounting packets to disk if the home server is down.
        #
#       Post-Proxy-Type Fail {
#                       detail
#       }
}
```

В следствии небольшой ошибки шаблонов конфигурационных файлов в файле dictionary будет содержаться не правильный путь на остальные словари. Необходимо исправить этот путь.

Пример файла /etc/raddb/dictionary:

```
#
#       This is the master dictionary file, which references the
#       pre-defined dictionary files included with the server.
#
#       Any new/changed attributes MUST be placed in this file, as
#       the pre-defined dictionaries SHOULD NOT be edited.
#
#       $Id$
#


#
#  The DHCP dictionary is used only when the server is built with
#  "configure --with-dhcp".  It is not (and should not) be used in
#  other situations.  If you are running just a RADIUS server, this
#  line can be deleted.  If you are using DHCP, the following line
#  should be uncommented.
#
#  Ideally, the "configure" process should automatically enable this
#  dictionary, but we don't yet do that.
#
#$INCLUDE       //dictionary.dhcp

#
#       The filename given here should be an absolute path.
#
$INCLUDE        /usr/share/freeradius/dictionary

#
#       Place additional attributes or $INCLUDEs here.  They will
#       over-ride the definitions in the pre-defined dictionaries.
#
#       See the 'man' page for 'dictionary' for information on
#       the format of the dictionary files.

#
#       If you want to add entries to the dictionary file,
#       which are NOT going to be placed in a RADIUS packet,
#       add them here.  The numbers you pick should be between
#       3000 and 4000.
#

#ATTRIBUTE      My-Local-String         3000    string
#ATTRIBUTE      My-Local-IPAddr         3001    ipaddr
#ATTRIBUTE      My-Local-Integer        3002    integer
```

2.3.9 Тестирование работы сервера FreeRADIUS

После установки сервера FreeRADIUS можно произвести тестирование его работы. Для тестирования и отладки работы сервера удобным является отладочный режим, при этом все сообщения сервера будут выводиться на консоль на которой сервер был запущен. Запуск сервера в отладочном режиме производиться следующей командой:

```
# radiusd -X
```

В ходе запуска сервера будет выводится сообщения о статусе загрузки сервера и его модулей. После загрузки на консоль будет выведено сообщение

"Ready to process requests." ("Готов к обработке запросов"). Отсутствие сообщения говорит о том что сервер не запустился, для выяснения причины сбоя необходимо рассмотреть выводимые сервером сообщения.

Тестирования сервера возможно с помощью скрипта radtest, который имеет следующую последовательность параметров запуска:

```
$ radtest user passwd radius-server[:port] nas-port-number secret
[ppphint] [nasname]
```

Значение secret устанавливается через конфигурационный файл clients.conf, в котором соотноситься адрес клиента который генерирует RADIUS запросы и секретного ключа, при помощи которого производится шифрование пакета.

Вызов данного скрипта для тестирования созданных в пункте 3 пользователей будет иметь следующий вид:

```
# radtest linux test123 127.0.0.1 0 testing123
```

В консоли где был запущен сервер должны быть выведены сообщения, что был получен запрос и на основе чего был сгенерирован ответ на данный запрос:

```
rad_recv: Access-Request packet from host 127.0.0.1 port 58278, id=38, length=75
        User-Name = "linux"
        User-Password = "test123"
        NAS-IP-Address = 10.100.2.10
        NAS-Port = 0
        Message-Authenticator = 0xe605727bd4887194c6415fb2cc254a1a
# Executing section authorize from file //etc/raddb/sites-enabled/default
+- entering group authorize {...}
++[preprocess] returns ok
++[chap] returns noop
++[mschap] returns noop
++[digest] returns noop
[suffix] No '@' in User-Name = "linux", looking up realm NULL
[suffix] No such realm "NULL"
++[suffix] returns noop
[eap] No EAP-Message, not doing EAP
++[eap] returns noop
++[files] returns noop
[sql]   expand: %{User-Name} -> linux
[sql] sql_set_user escaped user --> 'linux'
rlm_sql (sql): Reserving sql socket id: 4
[sql]   expand: SELECT id, UserName, Attribute, Value, Op   FROM radcheck   WHERE
Username = '%{SQL-User-Name}'   ORDER BY id -> SELECT id, UserName, Attribute, Value, Op   FROM
radcheck   WHERE Username = 'linux'   ORDER BY id
rlm_sql_postgresql: Status: PGRES_TUPLES_OK
rlm_sql_postgresql: query affected rows = 1 , fields = 5
WARNING: Found User-Password == "...".
WARNING: Are you sure you don't mean Cleartext-Password?
WARNING: See "man rlm_pap" for more information.
[sql] User found in radcheck table
[sql]   expand: SELECT id, UserName, Attribute, Value, Op   FROM radreply   WHERE
Username = '%{SQL-User-Name}'   ORDER BY id -> SELECT id, UserName, Attribute, Value, Op   FROM
radreply   WHERE Username = 'linux'   ORDER BY id
rlm_sql_postgresql: Status: PGRES_TUPLES_OK
rlm_sql_postgresql: query affected rows = 0 , fields = 5
[sql]   expand: SELECT GroupName FROM radusergroup WHERE UserName='%{SQL-User-Name}'
ORDER BY priority -> SELECT GroupName FROM radusergroup WHERE UserName='linux' ORDER BY priority
rlm_sql_postgresql: Status: PGRES_TUPLES_OK
rlm_sql_postgresql: query affected rows = 0 , fields = 1
rlm_sql (sql): Released sql socket id: 4
++[sql] returns ok
++[expiration] returns noop
++[logintime] returns noop
++[pap] returns updated
Found Auth-Type = PAP
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
!!!    Replacing User-Password in config items with Cleartext-Password.     !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Please update your configuration so that the "known good"               !!!
!!! clear text password is in Cleartext-Password, and not in User-Password. !!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
# Executing group from file //etc/raddb/sites-enabled/default
+- entering group PAP {...}
[pap] login attempt with password "test123"
[pap] Using clear text password "test123"
[pap] User authenticated successfully
++[pap] returns ok
# Executing section post-auth from file //etc/raddb/sites-enabled/default
+- entering group post-auth {...}
++[exec] returns noop
Sending Access-Accept of id 38 to 127.0.0.1 port 58278
Finished request 0.
Going to the next request
Waking up in 4.9 seconds.
Cleaning up request 0 ID 38 with timestamp +1539
Ready to process requests.
```

Программа radtest должна получить удовлетворительный ответ на запрос в течении нескольких секунд:

```
Sending Access-Request of id 38 to 127.0.0.1 port 1812
        User-Name = "linux"
        User-Password = "test123"
        NAS-IP-Address = 10.100.2.10
        NAS-Port = 0
        Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=38, length=20
```

2.3.10 Конфигурирование pptp сервера poptop

Конфигурирование PPTP сервера POPTOP было произведено в первой лабораторной работе. Для добавления возможности данному серверу производить аутентификацию пользователей не через файл а через БД, необходимо внести незначительные изменения в созданные в ходе первой лабораторной работы файлы и создать конфигурационные файлы RADIUS клиента – плагинов radius.so и plugin radattr.so

К изменениям которые необходимо внести в созданные в первой лабораторной работе файлы является добавление строк:

```
plugin radius.so
plugin radattr.so
```

в конец файла /etc/ppp/options.pptpd.

Необходимо установить клиентскую библиотеку radiusclient-ng (http://developer.berlios.de/projects/radiusclient-ng/). Выполним сборку и установку из исходных кодов:

```
$ tar xvf radiusclient-ng-<version>.tar.gz
$ cd radiusclient-ng-<version>
$ ./configure --prefix=/ --exec-prefix=/usr –datarootdir=/usr/share
$ make
$ sudo -s
# make install
```

Общий конфигурационный файл RADIUS клиента /etc/radiusclient-ng/radiusclient.conf:

```
# General settings

# specify which authentication comes first respectively which
# authentication is used. possible values are: "radius" and "local".
# if you specify "radius,local" then the RADIUS server is asked
```

```
# first then the local one. if only one keyword is specified only
# this server is asked.
auth_order      radius,local

# maximum login tries a user has
login_tries     4

# timeout for all login tries
# if this time is exceeded the user is kicked out
login_timeout   60

# name of the nologin file which when it exists disables logins.
# it may be extended by the ttyname which will result in
# a terminal specific lock (e.g. /etc/nologin.ttyS2 will disable
# logins on /dev/ttyS2)
nologin /etc/nologin

# name of the issue file. it's only display when no username is passed
# on the radlogin command line
issue   //etc/radiusclient-ng/issue

# RADIUS settings

# RADIUS server to use for authentication requests. this config
# item can appear more then one time. if multiple servers are
# defined they are tried in a round robin fashion if one
# server is not answering.
# optionally you can specify a the port number on which is remote
# RADIUS listens separated by a colon from the hostname. if
# no port is specified /etc/services is consulted of the radius
# service. if this fails also a compiled in default is used.
authserver      localhost

# RADIUS server to use for accouting requests. All that I
# said for authserver applies, too.
#
acctserver      localhost

# file holding shared secrets used for the communication
# between the RADIUS client and server
servers         //etc/radiusclient-ng/servers

# dictionary of allowed attributes and values
# just like in the normal RADIUS distributions
dictionary      //etc/radiusclient-ng/dictionary

# program to call for a RADIUS authenticated login
login_radius    /usr/sbin/login.radius

# file which holds sequence number for communication with the
# RADIUS server
seqfile         /var/run/radius.seq

# file which specifies mapping between ttyname and NAS-Port attribute
mapfile         //etc/radiusclient-ng/port-id-map

# default authentication realm to append to all usernames if no
# realm was explicitly specified by the user
# the radiusd directly form Livingston doesnt use any realms, so leave
# it blank then
default_realm

# time to wait for a reply from the RADIUS server
radius_timeout  10

# resend request this many times before trying the next server
radius_retries  3

# local address from which radius packets have to be sent
#bindaddr *

# LOCAL settings

# program to execute for local login
# it must support the -f flag for preauthenticated login
login_local     /bin/login
```

В интересах совместимости с legacy версиями RADIUS клиента необходимо создать символическую ссылку на каталог с конфигурационными файлами:

```
# ln -s /etc/radiusclient-ng/ /etc/radiusclient
```

Конфигурационный файл RADIUS клиента, с общими секретами данного клиента и RADIUS серверов с которыми он может взаимодействовать /etc/radiusclient-ng/servers (значение ключа должно совпадать со значение введённым в конфигурационном файле FreeRADIUS сервера – /etc/raddb/clients.conf):

```
#Server Name or Client/Server pair              Key
#-----------------                              ---------------
#portmaster.elemental.net                       hardlyasecret
#portmaster2.elemental.net                      donttellanyone
127.0.0.1                                       testing123
```

Также необходимо внести следующие изменения в файл /etc/raddb/modules/mschap сервера FreeRADIUS (соответствующие настройки были включены в первой лабораторной работе на сервере POPTOP в файле /etc/ppp/options.pptpd):

```
# -*- text -*-
#
#  $Id$

# Microsoft CHAP authentication
#
#  This module supports MS-CHAP and MS-CHAPv2 authentication.
#  It also enforces the SMB-Account-Ctrl attribute.
#
mschap {
        #
        #  If you are using /etc/smbpasswd, see the 'passwd'
        #  module for an example of how to use /etc/smbpasswd

        # if use_mppe is not set to no mschap will
        # add MS-CHAP-MPPE-Keys for MS-CHAPv1 and
        # MS-MPPE-Recv-Key/MS-MPPE-Send-Key for MS-CHAPv2
        #
        use_mppe = yes

        # if mppe is enabled require_encryption makes
        # encryption moderate
        #
        require_encryption = yes

        # require_strong always requires 128 bit key
        # encryption
        #
        require_strong = yes

        # Windows sends us a username in the form of
        # DOMAIN\user, but sends the challenge response
        # based on only the user portion.  This hack
        # corrects for that incorrect behavior.
        #
#       with_ntdomain_hack = no

        # The module can perform authentication itself, OR
        # use a Windows Domain Controller.  This configuration
        # directive tells the module to call the ntlm_auth
        # program, which will do the authentication, and return
        # the NT-Key.  Note that you MUST have "winbindd" and
        # "nmbd" running on the local machine for ntlm_auth
        # to work.  See the ntlm_auth program documentation
        # for details.
        #
```

```
                # If ntlm_auth is configured below, then the mschap
                # module will call ntlm_auth for every MS-CHAP
                # authentication request.  If there is a cleartext
                # or NT hashed password available, you can set
                # "MS-CHAP-Use-NTLM-Auth := No" in the control items,
                # and the mschap module will do the authentication itself,
                # without calling ntlm_auth.
                #
                # Be VERY careful when editing the following line!
                #
                # You can also try setting the user name as:
                #
                #       ... --username=%{mschap:User-Name} ...
                #
                # In that case, the mschap module will look at the User-Name
                # attribute, and do prefix/suffix checks in order to obtain
                # the "best" user name for the request.
                #
        #       ntlm_auth = "/path/to/ntlm_auth --request-nt-key --username=%{%{Stripped-User-
Name}:-%{%{User-Name}:-None}} --challenge=%{%{mschap:Challenge}:-00} --nt-response=%{%{mschap:NT-
Response}:-00}"

                # For Apple Server, when running on the same machine as
                # Open Directory.  It has no effect on other systems.
                #
        #       use_open_directory = yes

                # On failure, set (or not) the MS-CHAP error code saying
                # "retries allowed".
        #       allow_retry = yes

                # An optional retry message.
        #       retry_msg = "Re-enter (or reset) the password"
        }
```

Для корректного использования атрибутов Microsoft RADIUS клиенту нужен дополнительный словарь. Создайте файл /etc/radiusclient-ng/dictionary.microsoft со следующим содержанием:

```
        #
        #       Microsoft's VSA's, from RFC 2548
        #
        #       $Id: poptop_ads_howto_8.htm,v 1.8 2008/10/02 08:11:48 wskwok Exp $
        #
        VENDOR          Microsoft       311     Microsoft
        BEGIN VENDOR    Microsoft
        ATTRIBUTE       MS-CHAP-Response        1       string  Microsoft
        ATTRIBUTE       MS-CHAP-Error           2       string  Microsoft
        ATTRIBUTE       MS-CHAP-CPW-1           3       string  Microsoft
        ATTRIBUTE       MS-CHAP-CPW-2           4       string  Microsoft
        ATTRIBUTE       MS-CHAP-LM-Enc-PW       5       string  Microsoft
        ATTRIBUTE       MS-CHAP-NT-Enc-PW       6       string  Microsoft
        ATTRIBUTE       MS-MPPE-Encryption-Policy 7     string  Microsoft
        # This is referred to as both singular and plural in the RFC.
        # Plural seems to make more sense.
        ATTRIBUTE       MS-MPPE-Encryption-Type 8       string  Microsoft
        ATTRIBUTE       MS-MPPE-Encryption-Types  8     string  Microsoft
        ATTRIBUTE       MS-RAS-Vendor           9       integer Microsoft
        ATTRIBUTE       MS-CHAP-Domain          10      string  Microsoft
        ATTRIBUTE       MS-CHAP-Challenge       11      string  Microsoft
        ATTRIBUTE       MS-CHAP-MPPE-Keys       12      string  Microsoft encrypt=1
        ATTRIBUTE       MS-BAP-Usage            13      integer Microsoft
        ATTRIBUTE       MS-Link-Utilization-Threshold 14 integer        Microsoft
        ATTRIBUTE       MS-Link-Drop-Time-Limit 15      integer Microsoft
        ATTRIBUTE       MS-MPPE-Send-Key        16      string  Microsoft
        ATTRIBUTE       MS-MPPE-Recv-Key        17      string  Microsoft
        ATTRIBUTE       MS-RAS-Version          18      string  Microsoft
        ATTRIBUTE       MS-Old-ARAP-Password    19      string  Microsoft
        ATTRIBUTE       MS-New-ARAP-Password    20      string  Microsoft
        ATTRIBUTE       MS-ARAP-PW-Change-Reason 21     integer Microsoft
        ATTRIBUTE       MS-Filter               22      string  Microsoft
        ATTRIBUTE       MS-Acct-Auth-Type       23      integer Microsoft
        ATTRIBUTE       MS-Acct-EAP-Type        24      integer Microsoft
        ATTRIBUTE       MS-CHAP2-Response       25      string  Microsoft
        ATTRIBUTE       MS-CHAP2-Success        26      string  Microsoft
        ATTRIBUTE       MS-CHAP2-CPW            27      string  Microsoft
        ATTRIBUTE       MS-Primary-DNS-Server   28      ipaddr
        ATTRIBUTE       MS-Secondary-DNS-Server 29      ipaddr
```

```
ATTRIBUTE       MS-Primary-NBNS-Server  30      ipaddr Microsoft
ATTRIBUTE       MS-Secondary-NBNS-Server 31     ipaddr Microsoft
#ATTRIBUTE      MS-ARAP-Challenge       33      string  Microsoft
#
#       Integer Translations
#
#       MS-BAP-Usage Values
VALUE           MS-BAP-Usage            Not-Allowed     0
VALUE           MS-BAP-Usage            Allowed         1
VALUE           MS-BAP-Usage            Required        2
#       MS-ARAP-Password-Change-Reason Values
VALUE   MS-ARAP-PW-Change-Reason        Just-Change-Password            1
VALUE   MS-ARAP-PW-Change-Reason        Expired-Password                2
VALUE   MS-ARAP-PW-Change-Reason        Admin-Requires-Password-Change  3
VALUE   MS-ARAP-PW-Change-Reason        Password-Too-Short              4
#       MS-Acct-Auth-Type Values
VALUE           MS-Acct-Auth-Type       PAP             1
VALUE           MS-Acct-Auth-Type       CHAP            2
VALUE           MS-Acct-Auth-Type       MS-CHAP-1       3
VALUE           MS-Acct-Auth-Type       MS-CHAP-2       4
VALUE           MS-Acct-Auth-Type       EAP             5
#       MS-Acct-EAP-Type Values
VALUE           MS-Acct-EAP-Type        MD5             4
VALUE           MS-Acct-EAP-Type        OTP             5
VALUE           MS-Acct-EAP-Type        Generic-Token-Card      6
VALUE           MS-Acct-EAP-Type        TLS             13
END-VENDOR Microsoft
```

Подключите этот словарь, а также словарь /etc/radiusclient-ng/dictionary.merit, дописав в конец файла /etc/radiusclient-ng/dictionary следующие строки:

```
INCLUDE /etc/radiusclient-ng/dictionary.merit
INCLUDE /etc/radiusclient-ng/dictionary.microsoft
```

2.3.11 Проверка работы vpn клиента из ОС Windows и ОС Linux

Относительно клиентской стороны никаких изменений, относительно первой лабораторной работы, вносить ну нужно. Добавление механизма авторизации аутентификации и учёта посредством протокола RADIUS расширило удобства возможности администрирования клиентов и учёта их работы.

Для примера, так как сервер FreeRADIUS был настроен что бы учётные данные сохранялись в БД, можно посмотреть статистику работы клиента:

```
$ sudo -u postgres psql -h 127.0.0.1 freeradius freeradius
psql (9.1.1)
Type "help" for help.

freeradius=> select UserName, AcctStartTime, AcctStopTime, FramedProtocol from radacct;
 username |     acctstarttime      | acctstoptime | framedprotocol
----------+------------------------+--------------+----------------
 linux    | 2011-10-30 13:25:45-04 |              | PPP
(1 row)

freeradius=> \q
```

Логи серверов POPTOP и FreeRADIUS при удачной авторизации VPN клиента приведены ниже:

Лог сервера POPTOP:

```
Oct 30 13:31:48 serpentarium pptpd[4758]: MGR: Launching /usr/sbin/pptpctrl to handle
client
Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: local address = 192.168.0.234
Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: remote address = 192.168.1.234
Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: pppd options file =
/etc/ppp/options.pptpd
Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: Client 10.100.2.135 control connection
```

```
started
        Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: Received PPTP Control Message (type: 1)
        Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: Made a START CTRL CONN RPLY packet
        Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: I wrote 156 bytes to the client.
        Oct 30 13:31:48 serpentarium pptpd[4758]: CTRL: Sent packet to client
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: Received PPTP Control Message (type: 7)
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: Set parameters to 10000000 maxbps, 3
window size
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: Made a OUT CALL RPLY packet
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: Starting call (launching pppd, opening
GRE)
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: pty_fd = 6
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: tty_fd = 7
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: I wrote 32 bytes to the client.
        Oct 30 13:31:49 serpentarium pptpd[4758]: CTRL: Sent packet to client
        Oct 30 13:31:49 serpentarium pptpd[4759]: CTRL (PPPD Launcher): program binary =
/usr/sbin/pppd
        Oct 30 13:31:49 serpentarium pptpd[4759]: CTRL (PPPD Launcher): local address =
192.168.0.234
        Oct 30 13:31:49 serpentarium pptpd[4759]: CTRL (PPPD Launcher): remote address =
192.168.1.234
        Oct 30 13:31:49 serpentarium pppd[4759]: Plugin radius.so loaded.
        Oct 30 13:31:49 serpentarium pppd[4759]: RADIUS plugin initialized.
        Oct 30 13:31:49 serpentarium pppd[4759]: Plugin radattr.so loaded.
        Oct 30 13:31:49 serpentarium pppd[4759]: RADATTR plugin initialized.
        Oct 30 13:31:49 serpentarium pppd[4759]: Plugin /usr/lib/pptpd/pptpd-logwtmp.so loaded.
        Oct 30 13:31:49 serpentarium pppd[4759]: pppd 2.4.5 started by root, uid 0
        Oct 30 13:31:49 serpentarium pppd[4759]: Using interface ppp0
        Oct 30 13:31:49 serpentarium pppd[4759]: Connect: ppp0 <--> /dev/pts/2
        Oct 30 13:31:49 serpentarium pptpd[4758]: GRE: buffering packet #1 (expecting #0, lost or
reordered)
        Oct 30 13:31:49 serpentarium pptpd[4758]: GRE: buffering packet #2 (expecting #0, lost or
reordered)
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: timeout waiting for 1 packets
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting #1 from queue
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting #2 from queue
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #3
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #4
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #5
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #6
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #7
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #8
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #9
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #10
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #11
        Oct 30 13:31:52 serpentarium pppd[4759]: MPPE 128-bit stateless compression enabled
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #12
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #13
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #14
        Oct 30 13:31:52 serpentarium pptpd[4758]: GRE: accepting packet #15
        Oct 30 13:31:52 serpentarium pppd[4759]: Cannot determine ethernet address for proxy ARP
        Oct 30 13:31:52 serpentarium pppd[4759]: local  IP address 192.168.0.234
        Oct 30 13:31:52 serpentarium pppd[4759]: remote IP address 192.168.1.234
```

Лог сервера FreeRADIUS:

```
        rad_recv: Access-Request packet from host 127.0.0.1 port 38527, id=50, length=147
                Service-Type = Framed-User
                Framed-Protocol = PPP
                User-Name = "linux"
                MS-CHAP-Challenge = 0x3651751a0b084636e6bb96013e80b1ce
                MS-CHAP2-Response =
0x88003197ced72bd4b6f221ac0a40a4863c2400000000000000002e1dd371410e2e27f224b2b7180b4bf9cb9d147b1ec
19da3
                Calling-Station-Id = "10.100.2.135"
                NAS-IP-Address = 10.100.2.10
                NAS-Port = 0
        # Executing section authorize from file //etc/raddb/sites-enabled/default
        +- entering group authorize {...}
        ++[preprocess] returns ok
        ++[chap] returns noop
        [mschap] Found MS-CHAP attributes.  Setting 'Auth-Type  = mschap'
        ++[mschap] returns ok
        ++[digest] returns noop
        [suffix] No '@' in User-Name = "linux", looking up realm NULL
        [suffix] No such realm "NULL"
        ++[suffix] returns noop
        [eap] No EAP-Message, not doing EAP
        ++[eap] returns noop
```

```
        [files] users: Matched entry DEFAULT at line 172
        ++[files] returns ok
        [sql]   expand: %{User-Name} -> linux
        [sql] sql_set_user escaped user --> 'linux'
        rlm_sql (sql): Reserving sql socket id: 1
        [sql]   expand: SELECT id, UserName, Attribute, Value, Op   FROM radcheck   WHERE
Username = '%{SQL-User-Name}'   ORDER BY id -> SELECT id, UserName, Attribute, Value, Op   FROM
radcheck   WHERE Username = 'linux'   ORDER BY id
        rlm_sql_postgresql: Status: PGRES_TUPLES_OK
        rlm_sql_postgresql: query affected rows = 1 , fields = 5
        WARNING: Found User-Password == "...".
        WARNING: Are you sure you don't mean Cleartext-Password?
        WARNING: See "man rlm_pap" for more information.
        [sql] User found in radcheck table
        [sql]   expand: SELECT id, UserName, Attribute, Value, Op   FROM radreply   WHERE
Username = '%{SQL-User-Name}'   ORDER BY id -> SELECT id, UserName, Attribute, Value, Op   FROM
radreply   WHERE Username = 'linux'   ORDER BY id
        rlm_sql_postgresql: Status: PGRES_TUPLES_OK
        rlm_sql_postgresql: query affected rows = 0 , fields = 5
        [sql]   expand: SELECT GroupName FROM radusergroup WHERE UserName='%{SQL-User-Name}'
ORDER BY priority -> SELECT GroupName FROM radusergroup WHERE UserName='linux' ORDER BY priority
        rlm_sql_postgresql: Status: PGRES_TUPLES_OK
        rlm_sql_postgresql: query affected rows = 0 , fields = 1
        rlm_sql (sql): Released sql socket id: 1
        ++[sql] returns ok
        ++[expiration] returns noop
        ++[logintime] returns noop
        [pap] WARNING: Auth-Type already set.  Not setting to PAP
        ++[pap] returns noop
        Found Auth-Type = MSCHAP
        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        !!!    Replacing User-Password in config items with Cleartext-Password.     !!!
        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        !!! Please update your configuration so that the "known good"               !!!
        !!! clear text password is in Cleartext-Password, and not in User-Password. !!!
        !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        # Executing group from file //etc/raddb/sites-enabled/default
        +- entering group MS-CHAP {...}
        [mschap] Creating challenge hash with username: linux
        [mschap] Told to do MS-CHAPv2 for linux with NT-Password
        [mschap] adding MS-CHAPv2 MPPE keys
        ++[mschap] returns ok
        # Executing section post-auth from file //etc/raddb/sites-enabled/default
        +- entering group post-auth {...}
        ++[exec] returns noop
        Sending Access-Accept of id 50 to 127.0.0.1 port 38527
                Framed-Protocol = PPP
                Framed-Compression = Van-Jacobson-TCP-IP
                MS-CHAP2-Success =
0x88533d3941394542334536344344453536453035304331433543324432433637414239383742303036463 3
                MS-MPPE-Recv-Key = 0x76a08ea53026518cc06e7ad093c4e2aa
                MS-MPPE-Send-Key = 0x7efc0f029ddcae5335b1837820a0e34e
                MS-MPPE-Encryption-Policy = 0x00000002
                MS-MPPE-Encryption-Types = 0x00000004
        Finished request 3.
        Going to the next request
        Waking up in 4.9 seconds.
        rad_recv: Accounting-Request packet from host 127.0.0.1 port 54319, id=51, length=111
                Acct-Session-Id = "4EAD8A08129700"
                User-Name = "linux"
                Acct-Status-Type = Start
                Service-Type = Framed-User
                Framed-Protocol = PPP
                Calling-Station-Id = "10.100.2.135"
                Acct-Authentic = RADIUS
                NAS-Port-Type = Async
                Framed-IP-Address = 192.168.1.234
                NAS-IP-Address = 10.100.2.10
                NAS-Port = 0
                Acct-Delay-Time = 0
        # Executing section preacct from file //etc/raddb/sites-enabled/default
        +- entering group preacct {...}
        ++[preprocess] returns ok
        [acct_unique] Hashing 'NAS-Port = 0,Client-IP-Address = 127.0.0.1,NAS-IP-Address =
10.100.2.10,Acct-Session-Id = "4EAD8A08129700",User-Name = "linux"'
        [acct_unique] Acct-Unique-Session-ID = "d8d1fa7eed249ecc".
        ++[acct_unique] returns ok
        [suffix] No '@' in User-Name = "linux", looking up realm NULL
        [suffix] No such realm "NULL"
        ++[suffix] returns noop
```

```
        ++[files] returns noop
        # Executing section accounting from file //etc/raddb/sites-enabled/default
        +- entering group accounting {...}
        [detail]        expand: %{Packet-Src-IP-Address} -> 127.0.0.1
        [detail]        expand: //var/log/radius/radacct/%{%{Packet-Src-IP-Address}:-%{Packet-
Src-IPv6-Address}}/detail-%Y%m%d -> //var/log/radius/radacct/127.0.0.1/detail-20111030
        [detail] //var/log/radius/radacct/%{%{Packet-Src-IP-Address}:-%{Packet-Src-IPv6-
Address}}/detail-%Y%m%d expands to //var/log/radius/radacct/127.0.0.1/detail-20111030
        [detail]        expand: %t -> Sun Oct 30 13:31:52 2011
        ++[detail] returns ok
        ++[unix] returns ok
        [radutmp]       expand: //var/log/radius/radutmp -> //var/log/radius/radutmp
        [radutmp]       expand: %{User-Name} -> linux
        ++[radutmp] returns ok
        [sql]   expand: %{User-Name} -> linux
        [sql] sql_set_user escaped user --> 'linux'
        [sql]   expand: %{NAS-Port} -> 0
        [sql]   expand: %{Acct-Delay-Time} -> 0
        [sql]   expand: INSERT INTO radacct   (AcctSessionId, AcctUniqueId, UserName, Realm,
NASIPAddress,    NASPortId, NASPortType, AcctStartTime, AcctAuthentic,   ConnectInfo_start,
CalledStationId, CallingStationId, ServiceType,   FramedProtocol, FramedIPAddress,
AcctStartDelay, XAscendSessionSvrKey)   VALUES('%{Acct-Session-Id}',   '%{Acct-Unique-Session-
Id}',   '%{SQL-User-Name}',   NULLIF('%{Realm}', ''),   '%{NAS-IP-Address}',   %{%{NAS-Port}:-
NULL},   '%{NAS-Port-Type}',   ('%S'::timestamp - '%{%{Acct-Delay-Time}:-0}'::interval),
'%{Acct-Authentic}',   '%{Connect-Info}',   '%{Called-Station-Id}',   '%{Calling-Station-Id}',
'%{Service-Type}',   '%{Framed-Protocol}',   NULLIF('%{Framed-IP-Address}', '')::inet,   0,
'%{X-Ascend-Session-Svr-Key}') -> INSERT INTO radacct   (AcctSessionId, AcctUniqueId, UserName,
Realm, NASIPAddress,    NASPortId, NASPortType, AcctStartTime, AcctAuthentic,
ConnectInfo_start, CalledStationId, CallingStationId, ServiceType,   FramedProtocol,
FramedIPAddress, AcctStartDelay,
        rlm_sql (sql): Reserving sql socket id: 0
        rlm_sql_postgresql: Status: PGRES_COMMAND_OK
        rlm_sql_postgresql: query affected rows = 1
        rlm_sql (sql): Released sql socket id: 0
        ++[sql] returns ok
        ++[exec] returns noop
        [attr_filter.accounting_response]       expand: %{User-Name} -> linux
        attr_filter: Matched entry DEFAULT at line 12
        ++[attr_filter.accounting_response] returns updated
        Sending Accounting-Response of id 51 to 127.0.0.1 port 54319
        Finished request 4.
        Cleaning up request 4 ID 51 with timestamp +581
        Going to the next request
        Waking up in 4.8 seconds.
        Cleaning up request 3 ID 50 with timestamp +581
        Ready to process requests.
```

Для облегчения администрирования клиентов и просмотра статистики можно написать программы, скрипты, web-интерфейсы и так далее. Однако на данный момент уже существует много разнообразных программ обеспечивающих выполнение данных задач в связке POPTOP и FreeRADIUS и другое ПО, например FreeNIBS.

2.4 Содержание отчета

Отчёт должен содержать ход выполнения лабораторной работы с листингами лог-файлов работы серверов POPTOP и FreeRADIUS, а также код SQL запросов производимых к БД.

2.5 Контрольные вопросы

1. Какие протоколы аутентификации, авторизации и учёта вы знаете?
2. Какие достоинства и недостатки протокола RADIUS?
3. К какому уровню модели OSI следует отнести протокол RADIUS? Какой транспорт он использует и почему?
4. Какой принцип аутентификации, авторизации и учёта в протоколе RADIUS? Какие команды запросов/ответов для этого применяются?

5. Сервер FreeRADIUS и его модули. Опишите алгоритм отработки RADIUS запроса данным сервером.
6. Какие модули сервера FreeRADIUS были использованы в данной лабораторной работе? Каково из назначение?
7. В чём преимущество использования БД в сервере FreeRADIUS? Какие ещё источники данных могу быть использованы?
8. Перечислите протоколы авторизации, которые поддерживаются сервером FreeRADIUS.
9. Особенности использования сервера FreeRADIUS для построения систем учета трафика VPN пользователей.
10. Возможен ли обмен аутентификационными данными пользователей (роуминг) между различными RADIUS серверами? Понятие областей (Realm), применение областей при организации роуминга.

3 Самостоятельная работа №3. Настройка почтового сервера postfix

3.1 Цель работы

На примере MTA postfix изучить конфигурирование почтового сервера в среде UNIX.

3.2 Краткие теоретические сведения

Postfix — агент передачи почты (MTA — mail transfer agent). Postfix является свободным программным обеспечением.

Postfix – это агент передачи сообщений (MTA, message transport agent), который занимается пересылкой по протоколу SMTP сообщений от пользовательского почтового агента (MUA, mail user agent), называемого также почтовым клиентом, к удаленному почтовому серверу.

MTA также принимает сообщения от удаленных почтовых серверов и пересылает их другим MTA или доставляет в локальные почтовые ящики. Переслав или доставив сообщение, Postfix заканчивает свою работу. За доставку сообщения конечному пользователю отвечают другие серверы. Например, такие MTA, как серверы POP3 или IMAP1, передают сообщения почтовым клиентам – Mutt, Outlook или Apple Mail, с помощью которых пользователь может прочитать их.

Postfix создавался как альтернатива Sendmail. Считается, что Postfix быстрее работает, легче в администрировании, более защищён и, что важно, совместим с Sendmail.

Изначально Postfix был разработан Вейтсом Венемой в то время, когда он работал в Исследовательском центре имени Томаса Уотсона компании IBM. Первые версии программы стали доступны в середине 1999 года.

Postfix отличается продуманной модульной архитектурой, которая позволяет создать очень надёжную и быструю почтовую систему. Так, например, привилегии root требуются только для открытия порта (TCP 25 порт), а демоны, которые выполняют основную работу, могут работать непривилегированным пользователем в изолированном (chroot) окружении, что очень положительно сказывается на безопасности.

Архитектура Postfix выполнена в стиле UNIX — где простые программы выполняют минимальный набор функций, но выполняют их быстро и надежно. При простое почтовой системы ненужные демоны могут прекращать свою работу, высвобождая тем самым память, а при необходимости снова запускаются master-демоном.

Также стоит отметить более простую и понятную конфигурацию по сравнению с Sendmail и меньшую ресурсоёмкость, особенно во время простоя почтовой системы.

Совместим с AIX, BSD, HP-UX, IRIX, GNU/Linux, Mac OS X, Solaris, Tru64 UNIX, фактически может быть собран на любой Unix-подобной

операционной системе, поддерживающей POSIX и имеющей компилятор C. Является службой доставки почты по умолчанию в ОС NetBSD.

3.3 Ход работы

3.3.1 Установка необходимых пакетов

Пакеты необходимые для лабораторной работы можно загрузить с их официальных сайтов:

- postfix (http://www.postfix.org/download.html);
- clamav (http://www.clamav.net/lang/en/download/sources/);
- spamassassin (http://spamassassin.apache.org/);
- postgres, который был установлен и настроен в прошлой лабораторной работе.

3.3.2 Установка пакета postfix

Перед установкой необходимо остановить и удалить из системы все установленные MTA (например sendmail или exim)!

Необходимо убедиться в наличии development версий библиотек libdb, libpq (libdb-devel, libpqxx-devel для Fedora, RHEL; libdb-dev, libpqxx-dev для Debian, Ubuntu). После make tidy необходимо указать опции сборки модуля взаимодействия с postgres (-DHAS_PGSQL), а также каталог с заголовочными файлами postgres (в данном случае /usr/include/postgresql), каталог с библиотеками (/usr/lib) и опцию линковки с libpq (-lpq); на этапе make install на все вопросы конфигуратора отвечать нажатием Enter).

Выполните сборку и установку postfix:

```
$ tar xvf postfix-<version>.tar.gz
$ cd ./postfix-<version>
$ make tidy
$ make -f Makefile.init makefiles CCARGS='-DHAS_PGSQL -
I/usr/include/postgresql' AUXLIBS='-L/usr/lib -lpq'
$ make
$ sudo -s
# useradd -d /dev/null -s /bin/false -M postfix
# student@serpentarium:~$ sudo groupadd postdrop
# make install
```

3.3.3 Настройка postfix

Postfix поддерживает несколько режимов работы: конечный сервер с UNIX аккаунтами (canonical_domain), конечный сервер с виртуальными хостами и не UNIX аккаунтами (также с хранением писем локально или с пересылкой на другой сервер), временный SMTP сервер, для хранения почты, которую отправили на электронный адрес, указанный в записях MX DNS.

В данной лабораторной работе рассматривается конфигурирование postfix для работы с виртуальными хостами, как наиболее гибкий и масштабируемый.

Сперва проведите базовую настройку сервера для роли локального MTA и standalone internet server (настройки сервера хранятся в файле /etc/postfix/main.cf, настройки программных каналов передачи писем в /etc/postfix/master.cf):

Пример конфигурационного файла /etc/postfix/main.cf:

```
# Global Postfix configuration file. This file lists only a subset
# of all parameters. For the syntax, and for a complete parameter
# list, see the postconf(5) manual page (command: "man 5 postconf").
#
# For common configuration examples, see BASIC_CONFIGURATION_README
# and STANDARD_CONFIGURATION_README. To find these documents, use
# the command "postconf html_directory readme_directory", or go to
# http://www.postfix.org/.
#
# For best results, change no more than 2-3 parameters at a time,
# and test if Postfix still works after every change.

# SOFT BOUNCE
#
# The soft_bounce parameter provides a limited safety net for
# testing.  When soft_bounce is enabled, mail will remain queued that
# would otherwise bounce. This parameter disables locally-generated
# bounces, and prevents the SMTP server from rejecting mail permanently
# (by changing 5xx replies into 4xx replies). However, soft_bounce
# is no cure for address rewriting mistakes or mail routing mistakes.
#
#soft_bounce = no

# LOCAL PATHNAME INFORMATION
#
# The queue_directory specifies the location of the Postfix queue.
# This is also the root directory of Postfix daemons that run chrooted.
# See the files in examples/chroot-setup for setting up Postfix chroot
# environments on different UNIX systems.
#
queue_directory = /var/spool/postfix

# The command_directory parameter specifies the location of all
# postXXX commands.
#
command_directory = /usr/sbin

# The daemon_directory parameter specifies the location of all Postfix
# daemon programs (i.e. programs listed in the master.cf file). This
# directory must be owned by root.
#
daemon_directory = /usr/libexec/postfix

# The data_directory parameter specifies the location of Postfix-writable
# data files (caches, random numbers). This directory must be owned
# by the mail_owner account (see below).
#
data_directory = /var/lib/postfix

# QUEUE AND PROCESS OWNERSHIP
#
# The mail_owner parameter specifies the owner of the Postfix queue
# and of most Postfix daemon processes.  Specify the name of a user
# account THAT DOES NOT SHARE ITS USER OR GROUP ID WITH OTHER ACCOUNTS
# AND THAT OWNS NO OTHER FILES OR PROCESSES ON THE SYSTEM.  In
# particular, don't specify nobody or daemon. PLEASE USE A DEDICATED
# USER.
#
mail_owner = postfix

# The default_privs parameter specifies the default rights used by
# the local delivery agent for delivery to external file or command.
# These rights are used in the absence of a recipient user context.
# DO NOT SPECIFY A PRIVILEGED USER OR THE POSTFIX OWNER.
#
#default_privs = nobody

# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
```

```
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
myhostname = localhost

# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# $mydomain is used as a default value for many other configuration
# parameters.
#
#mydomain = domain.tld

# SENDING MAIL
#
# The myorigin parameter specifies the domain that locally-posted
# mail appears to come from. The default is to append $myhostname,
# which is fine for small sites.  If you run a domain with multiple
# machines, you should (1) change this to $mydomain and (2) set up
# a domain-wide alias database that aliases each user to
# user@that.users.mailhost.
#
# For the sake of consistency between sender and recipient addresses,
# myorigin also specifies the default domain name that is appended
# to recipient addresses that have no @domain part.
#
myorigin = $myhostname
#myorigin = $mydomain

# RECEIVING MAIL

# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on.  By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
inet_interfaces = all
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost

# The proxy_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on by way of a
# proxy or network address translation unit. This setting extends
# the address list specified with the inet_interfaces parameter.
#
# You must specify your proxy/NAT addresses when your system is a
# backup MX host for other domains, otherwise mail delivery loops
# will happen when the primary MX host is down.
#
#proxy_interfaces =
#proxy_interfaces = 1.2.3.4

# The mydestination parameter specifies the list of domains that this
# machine considers itself the final destination for.
#
# These domains are routed to the delivery agent specified with the
# local_transport parameter setting. By default, that is the UNIX
# compatible delivery agent that lookups all recipients in /etc/passwd
# and /etc/aliases or their equivalent.
#
# The default is $myhostname + localhost.$mydomain.  On a mail domain
# gateway, you should also include $mydomain.
#
# Do not specify the names of virtual domains - those domains are
# specified elsewhere (see VIRTUAL_README).
#
# Do not specify the names of domains that this machine is backup MX
# host for. Specify those names via the relay_domains settings for
# the SMTP server, or use permit_mx_backup if you are lazy (see
# STANDARD_CONFIGURATION_README).
#
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
```

```
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
mydestination = $myhostname, localhost.$mydomain, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$mydomain, www.$mydomain, ftp.$mydomain

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
#
# The local_recipient_maps parameter specifies optional lookup tables
# with all names or addresses of users that are local with respect
# to $mydestination, $inet_interfaces or $proxy_interfaces.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown local users. This parameter is defined by default.
#
# To turn off local recipient checking in the SMTP server, specify
# local_recipient_maps = (i.e. empty).
#
# The default setting assumes that you use the default Postfix local
# delivery agent for local delivery. You need to update the
# local_recipient_maps setting if:
#
# - You define $mydestination domain recipients in files other than
#   /etc/passwd, /etc/aliases, or the $virtual_alias_maps files.
#   For example, you define $mydestination domain recipients in
#   the $virtual_mailbox_maps files.
#
# - You redefine the local delivery agent in master.cf.
#
# - You redefine the "local_transport" setting in main.cf.
#
# - You use the "luser_relay", "mailbox_transport", or "fallback_transport"
#   feature of the Postfix local delivery agent (see local(8)).
#
# Details are described in the LOCAL_RECIPIENT_README file.
#
# Beware: if the Postfix SMTP server runs chrooted, you probably have
# to access the passwd file via the proxymap service, in order to
# overcome chroot restrictions. The alternative, having a copy of
# the system passwd file in the chroot jail is just not practical.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify a bare username, an @domain.tld
# wild-card, or specify a user@domain.tld address.
#
#local_recipient_maps = unix:passwd.byname $alias_maps
#local_recipient_maps = proxy:unix:passwd.byname $alias_maps
#local_recipient_maps =

# The unknown_local_recipient_reject_code specifies the SMTP server
# response code when a recipient domain matches $mydestination or
# ${proxy,inet}_interfaces, while $local_recipient_maps is non-empty
# and the recipient address or address local-part is not found.
#
# The default setting is 550 (reject mail) but it is safer to start
# with 450 (try again later) until you are certain that your
# local_recipient_maps settings are OK.
#
unknown_local_recipient_reject_code = 550

# TRUST AND RELAY CONTROL

# The mynetworks parameter specifies the list of "trusted" SMTP
# clients that have more privileges than "strangers".
#
# In particular, "trusted" SMTP clients are allowed to relay mail
# through Postfix.  See the smtpd_recipient_restrictions parameter
# in postconf(5).
#
# You can specify the list of "trusted" network addresses by hand
# or you can let Postfix do it for you (which is the default).
#
```

```
# By default (mynetworks_style = subnet), Postfix "trusts" SMTP
# clients in the same IP subnetworks as the local machine.
# On Linux, this does works correctly only with interfaces specified
# with the "ifconfig" command.
#
# Specify "mynetworks_style = class" when Postfix should "trust" SMTP
# clients in the same IP class A/B/C networks as the local machine.
# Don't do this with a dialup site - it would cause Postfix to "trust"
# your entire provider's network.  Instead, specify an explicit
# mynetworks list by hand, as described below.
#
# Specify "mynetworks_style = host" when Postfix should "trust"
# only the local machine.
#
#mynetworks_style = class
mynetworks_style = subnet
#mynetworks_style = host

# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
mynetworks = 192.168.0.0/16, 10.100.0.0/16
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table

# The relay_domains parameter restricts what destinations this system will
# relay mail to.  See the smtpd_recipient_restrictions description in
# postconf(5) for detailed information.
#
# By default, Postfix relays mail
# - from "trusted" clients (IP address matches $mynetworks) to any destination,
# - from "untrusted" clients to destinations that match $relay_domains or
#   subdomains thereof, except addresses with sender-specified routing.
# The default relay_domains value is $mydestination.
#
# In addition to the above, the Postfix SMTP server by default accepts mail
# that Postfix is final destination for:
# - destinations that match $inet_interfaces or $proxy_interfaces,
# - destinations that match $mydestination
# - destinations that match $virtual_alias_domains,
# - destinations that match $virtual_mailbox_domains.
# These destinations do not need to be listed in $relay_domains.
#
# Specify a list of hosts or domains, /file/name patterns or type:name
# lookup tables, separated by commas and/or whitespace.  Continue
# long lines by starting the next line with whitespace. A file name
# is replaced by its contents; a type:name table is matched when a
# (parent) domain appears as lookup key.
#
# NOTE: Postfix will not automatically forward mail for domains that
# list this system as their primary or backup MX host. See the
# permit_mx_backup restriction description in postconf(5).
#
#relay_domains = $mydestination

# INTERNET OR INTRANET

# The relayhost parameter specifies the default host to send mail to
# when no entry is matched in the optional transport(5) table. When
# no relayhost is given, mail is routed directly to the destination.
#
# On an intranet, specify the organizational domain name. If your
# internal DNS uses no MX records, specify the name of the intranet
# gateway host instead.
#
# In the case of SMTP, specify a domain, host, host:port, [host]:port,
# [address] or [address]:port; the form [host] turns off MX lookups.
#
# If you're connected via UUCP, see also the default_transport parameter.
#
#relayhost = $mydomain
#relayhost = [gateway.my.domain]
```

```
#relayhost = [mailserver.isp.tld]
#relayhost = uucphost
relayhost =

# REJECTING UNKNOWN RELAY USERS
#
# The relay_recipient_maps parameter specifies optional lookup tables
# with all addresses in the domains that match $relay_domains.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown relay users. This feature is off by default.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify an @domain.tld wild-card, or specify
# a user@domain.tld address.
#
#relay_recipient_maps = hash:/etc/postfix/relay_recipients

# INPUT RATE CONTROL
#
# The in_flow_delay configuration parameter implements mail input
# flow control. This feature is turned on by default, although it
# still needs further development (it's disabled on SCO UNIX due
# to an SCO bug).
#
# A Postfix process will pause for $in_flow_delay seconds before
# accepting a new message, when the message arrival rate exceeds the
# message delivery rate. With the default 100 SMTP server process
# limit, this limits the mail inflow to 100 messages a second more
# than the number of messages delivered per second.
#
# Specify 0 to disable the feature. Valid delays are 0..10.
#
#in_flow_delay = 1s

# ADDRESS REWRITING
#
# The ADDRESS_REWRITING_README document gives information about
# address masquerading or other forms of address rewriting including
# username->Firstname.Lastname mapping.

# ADDRESS REDIRECTION (VIRTUAL DOMAIN)
#
# The VIRTUAL_README document gives information about the many forms
# of domain hosting that Postfix supports.

# "USER HAS MOVED" BOUNCE MESSAGES
#
# See the discussion in the ADDRESS_REWRITING_README document.

# TRANSPORT MAP
#
# See the discussion in the ADDRESS_REWRITING_README document.

# ALIAS DATABASE
#
# The alias_maps parameter specifies the list of alias databases used
# by the local delivery agent. The default list is system dependent.
#
# On systems with NIS, the default is to search the local alias
# database, then the NIS alias database. See aliases(5) for syntax
# details.
#
# If you change the alias database, run "postalias /etc/aliases" (or
# wherever your system stores the mail alias file), or simply run
# "newaliases" to build the necessary DBM or DB file.
#
# It will take a minute or so before changes become visible.  Use
# "postfix reload" to eliminate the delay.
#
#alias_maps = dbm:/etc/aliases
alias_maps = hash:/etc/aliases
#alias_maps = hash:/etc/aliases, nis:mail.aliases
#alias_maps = netinfo:/aliases

# The alias_database parameter specifies the alias database(s) that
# are built with "newaliases" or "sendmail -bi".  This is a separate
# configuration parameter, because alias_maps (see above) may specify
# tables that are not necessarily all under control by Postfix.
#
```

```
#alias_database = dbm:/etc/aliases
#alias_database = dbm:/etc/mail/aliases
alias_database = hash:/etc/aliases
#alias_database = hash:/etc/aliases, hash:/opt/majordomo/aliases

# ADDRESS EXTENSIONS (e.g., user+foo)
#
# The recipient_delimiter parameter specifies the separator between
# user names and address extensions (user+foo). See canonical(5),
# local(8), relocated(5) and virtual(5) for the effects this has on
# aliases, canonical, virtual, relocated and .forward file lookups.
# Basically, the software tries user+foo and .forward+foo before
# trying user and .forward.
#
#recipient_delimiter = +

# DELIVERY TO MAILBOX
#
# The home_mailbox parameter specifies the optional pathname of a
# mailbox file relative to a user's home directory. The default
# mailbox file is /var/spool/mail/user or /var/mail/user.  Specify
# "Maildir/" for qmail-style delivery (the / is required).
#
#home_mailbox = Mailbox
#home_mailbox = Maildir/

# The mail_spool_directory parameter specifies the directory where
# UNIX-style mailboxes are kept. The default setting depends on the
# system type.
#
#mail_spool_directory = /var/mail
#mail_spool_directory = /var/spool/mail

# The mailbox_command parameter specifies the optional external
# command to use instead of mailbox delivery. The command is run as
# the recipient with proper HOME, SHELL and LOGNAME environment settings.
# Exception:  delivery for root is done as $default_user.
#
# Other environment variables of interest: USER (recipient username),
# EXTENSION (address extension), DOMAIN (domain part of address),
# and LOCAL (the address localpart).
#
# Unlike other Postfix configuration parameters, the mailbox_command
# parameter is not subjected to $parameter substitutions. This is to
# make it easier to specify shell syntax (see example below).
#
# Avoid shell meta characters because they will force Postfix to run
# an expensive shell process. Procmail alone is expensive enough.
#
# IF YOU USE THIS TO DELIVER MAIL SYSTEM-WIDE, YOU MUST SET UP AN
# ALIAS THAT FORWARDS MAIL FOR ROOT TO A REAL USER.
#
#mailbox_command = /some/where/procmail
#mailbox_command = /some/where/procmail -a "$EXTENSION"

# The mailbox_transport specifies the optional transport in master.cf
# to use after processing aliases and .forward files. This parameter
# has precedence over the mailbox_command, fallback_transport and
# luser_relay parameters.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#mailbox_transport = lmtp:unix:/file/name
#mailbox_transport = cyrus

# The fallback_transport specifies the optional transport in master.cf
# to use for recipients that are not found in the UNIX passwd database.
# This parameter has precedence over the luser_relay parameter.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
```

```
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#fallback_transport = lmtp:unix:/file/name
#fallback_transport = cyrus
#fallback_transport =

# The luser_relay parameter specifies an optional destination address
# for unknown recipients.  By default, mail for unknown@$mydestination,
# unknown@[$inet_interfaces] or unknown@[$proxy_interfaces] is returned
# as undeliverable.
#
# The following expansions are done on luser_relay: $user (recipient
# username), $shell (recipient shell), $home (recipient home directory),
# $recipient (full recipient address), $extension (recipient address
# extension), $domain (recipient domain), $local (entire recipient
# localpart), $recipient_delimiter. Specify ${name?value} or
# ${name:value} to expand value only when $name does (does not) exist.
#
# luser_relay works only for the default Postfix local delivery agent.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must specify "local_recipient_maps =" (i.e. empty) in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#luser_relay = $user@other.host
#luser_relay = $local@other.host
#luser_relay = admin+$local

# JUNK MAIL CONTROLS
#
# The controls listed here are only a very small subset. The file
# SMTPD_ACCESS_README provides an overview.

# The header_checks parameter specifies an optional table with patterns
# that each logical message header is matched against, including
# headers that span multiple physical lines.
#
# By default, these patterns also apply to MIME headers and to the
# headers of attached messages. With older Postfix versions, MIME and
# attached message headers were treated as body text.
#
# For details, see "man header_checks".
#
#header_checks = regexp:/etc/postfix/header_checks

# FAST ETRN SERVICE
#
# Postfix maintains per-destination logfiles with information about
# deferred mail, so that mail can be flushed quickly with the SMTP
# "ETRN domain.tld" command, or by executing "sendmail -qRdomain.tld".
# See the ETRN_README document for a detailed description.
#
# The fast_flush_domains parameter controls what destinations are
# eligible for this service. By default, they are all domains that
# this server is willing to relay mail to.
#
#fast_flush_domains = $relay_domains

# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
# You MUST specify $myhostname at the start of the text. That is an
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)

# PARALLEL DELIVERY TO THE SAME DESTINATION
#
# How many parallel deliveries to the same user or domain? With local
# delivery, it does not make sense to do massively parallel delivery
```

```
# to the same user, because mailbox updates must happen sequentially,
# and expensive pipelines in .forward files can cause disasters when
# too many are run at the same time. With SMTP deliveries, 10
# simultaneous connections to the same domain could be sufficient to
# raise eyebrows.
#
# Each message delivery transport has its XXX_destination_concurrency_limit
# parameter.  The default is $default_destination_concurrency_limit for
# most delivery transports. For the local delivery agent the default is 2.

#local_destination_concurrency_limit = 2
#default_destination_concurrency_limit = 20

# DEBUGGING CONTROL
#
# The debug_peer_level parameter specifies the increment in verbose
# logging level when an SMTP client or server host name or address
# matches a pattern in the debug_peer_list parameter.
#
debug_peer_level = 2

# The debug_peer_list parameter specifies an optional list of domain
# or network patterns, /file/name patterns or type:name tables. When
# an SMTP client or server host name or address matches a pattern,
# increase the verbose logging level by the amount specified in the
# debug_peer_level parameter.
#
#debug_peer_list = 127.0.0.1
#debug_peer_list = some.domain

# The debugger_command specifies the external command that is executed
# when a Postfix daemon program is run with the -D option.
#
# Use "command .. & sleep 5" so that the debugger can attach before
# the process marches on. If you use an X-based debugger, be sure to
# set up your XAUTHORITY environment variable before starting Postfix.
#
debugger_command =
         PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
         ddd $daemon_directory/$process_name $process_id & sleep 5

# If you can't use X, use this to capture the call stack when a
# daemon crashes. The result is in a file in the configuration
# directory, and is named after the process name and the process ID.
#
# debugger_command =
#       PATH=/bin:/usr/bin:/usr/local/bin; export PATH; (echo cont;
#       echo where) | gdb $daemon_directory/$process_name $process_id 2>&1
#       >$config_directory/$process_name.$process_id.log & sleep 5
#
# Another possibility is to run gdb under a detached screen session.
# To attach to the screen sesssion, su root and run "screen -r
# <id_string>" where <id_string> uniquely matches one of the detached
# sessions (from "screen -list").
#
# debugger_command =
#       PATH=/bin:/usr/bin:/sbin:/usr/sbin; export PATH; screen
#       -dmS $process_name gdb $daemon_directory/$process_name
#       $process_id & sleep 1

# INSTALL-TIME CONFIGURATION INFORMATION
#
# The following parameters are used when installing a new Postfix version.
#
# sendmail_path: The full pathname of the Postfix sendmail command.
# This is the Sendmail-compatible mail posting interface.
#
sendmail_path = /usr/sbin/sendmail

# newaliases_path: The full pathname of the Postfix newaliases command.
# This is the Sendmail-compatible command to build alias databases.
#
newaliases_path = /usr/bin/newaliases

# mailq_path: The full pathname of the Postfix mailq command.  This
# is the Sendmail-compatible mail queue listing command.
#
mailq_path = /usr/bin/mailq

# setgid_group: The group for mail submission and queue management
```

```
# commands.  This must be a group name with a numerical group ID that
# is not shared with other accounts, not even with the Postfix account.
#
setgid_group = postdrop

# html_directory: The location of the Postfix HTML documentation.
#
html_directory = no

# manpage_directory: The location of the Postfix on-line manual pages.
#
manpage_directory = /usr/local/man

# sample_directory: The location of the Postfix sample configuration files.
# This parameter is obsolete as of Postfix 2.1.
#
sample_directory = /etc/postfix

# readme_directory: The location of the Postfix README files.
#
readme_directory = no
```

В данном конфигурационном файле присутствуют:

- myorigin — доменное имя, которое будет использовано в качестве сервера отправки/получения локальных писем;
- mydestination — список конечный адресов (в данном случае почта может быть отправлена только нам, mta не будет пересылать почту, адресуемую чужим smtp серверам);
- mynetwork_style — указывает ограничение на возможных отправителей (в данном случае ограничиваем подсеть, откуда могут приниматься письма);
- mynetworks — подсети, с которых будет производится получение писем;
- relayhost — следующий узел для пересылки почты (пустое значение означает отправку писем непосредственно адресату, вместо пересылки на чужой smtp);
- alias_maps, alias_database — БД для задания псевдонимов локальных пользователей (оставлено по-умолчанию).

Остальные опции были оставлены не тронутыми. Настройка SSL в данной лабораторной работе не рассматривалась.

Добавьте виртуальный хост mail.lan. Отправка писем на виртуальный хост будет возможна адресатам, не имеющих UNIX аккаунтов на сервере (в отличии от, например, локальных пользователей). Для каждого пользователя будет выделен каталог для хранения писем, структура которого совместима с рядом pop3/imap серверов, что позволит развернуть полноценный email сервер.

Добавьте в конец конфигурационного файла /etc/postfix/main.cf следующие строки:

```
virtual_mailbox_domains = mail.lan
virtual_mailbox_maps = pgsql:/etc/postfix/pgsql-vmailbox.cf
virtual_mailbox_base = /var/mail/vhosts
virtual_gid_maps = static:5000
virtual_uid_maps = static:5000
virtual_minimum_uid = 1000
```

Данные строки обозначают:

- virtual_mailbox_domains — задает список виртуальных хостов;

– virtual_mailbox_maps — настройки соединения для подключения к БД с аккаунтами и сопоставленными каждому аккаунту каталогами хранения почты (или сама БД — текстовый файл, если указано, как hash:/path/to/db);
– virtual_mailbox_base родительский каталог для хранения писем;
– virtual_gid_maps, virtual_uid_maps — uid & gid папок с письмами (из соображений безопасности);
– virtual_minimum_uid — минимальное значение uid для папок с письмами.

Создайте необходимую иерархию каталогов для писем виртуальных пользователей и усановим владельца:

```
# groupadd --gid 5000 vmail
# useradd --uid 5000 -s /bin/false -d /dev/null -g vmail vmail
# mkdir -p /var/mail/vhosts/mail.lan/santa/
# mkdir -p /var/mail/vhosts/mail.lan/test/
# chown -R vmail:vmail /var/mail/vhosts/*
```

Далее создайте саму БД и роли к ней (пользователь postgres имеет адмиинстративные права и по-умолчанию не имеет пароля):

```
$ sudo -u postgres createuser -P vmail
Enter password for new role: vmail
Enter it again: vmail
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
$ sudo -u postgres createdb -O vmail postfix
$ sudo -u postgres psql -d postfix
psql (9.1.1)
Type "help" for help.

postfix=# create table vmailbox (name varchar, address varchar);
CREATE TABLE
postfix=# alter table vmailbox owner to vmail;
ALTER TABLE
postfix=# insert into vmailbox (name, address) values ('santa@mail.lan',
'mail.lan/santa/');
INSERT 0 1
postfix=# insert into vmailbox (name, address) values ('test@mail.lan',
'mail.lan/test/');
INSERT 0 1
postfix=# \q
```

По умолчанию проинициализированный репозиторий БД настроен так что аутентификация всех пользователей и БД производиться методом ident sameuser (соответствие пользователей PostgreSQL, владельца БД и пользователя ОС системы). Поэтому в конфигурационный файл методов аутентификации клиентов PosgtreSQL (по умолчанию /var/lib/pgsql/data/pg_hba.conf), необходимо перед всеми остальными настройками, добавить следующую строчку:

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
host    postfix         vmail           127.0.0.1/32            password
```

Задайте настройки подключения к БД для postfix.

Пример конфигурационного файла /etc/postfix/pgsql-vmailbox.cf:

```
hosts = 127.0.0.1

# The user name and password to log into the pgsql server.
user = vmail
password = vmail
```

```
# The database name on the servers.
dbname = postfix

# Postfix 2.2 and later The SQL query template. See pgsql_table(5).
query = SELECT address FROM vmailbox WHERE name='%s'
```

Запустите postfix:

```
# postalias /etc/aliases
# postfix start
```

Убедитесь в том, что демон напустился:

```
$ ps ax | grep postfix
23804 ?        Ss     0:00 /usr/libexec/postfix/master

$ netstat -l | grep smtp
tcp        0      0 *:smtp                  *:*                     LISTEN
```

Логи Postfix можно почитать в файлах /var/log/mail.err, /var/log/mail.info, /var/log/mail.log, /var/log/mail.warn.

Чтобы postfix перечитал конфигурацию применяются команда # postfix reload. Если необходимо оповестить posfix о новых файлах с алиасами (не настройками, а непосредственно алиасами, не важно чего — postfix много чего может алиасить): # postalias /path/to/aliases/file, для указания файла с алиасами: # postmap /path/to/aliases/file.

3.3.4 Проверка заданных настроек postfix

Создайте нового системного пользователя, которому будет отправлено тестовое письмо через локальную рассылку:

```
# useradd testuser
```

Проверьте работоспособность локальной рассылки:

```
$ mail -s "Local mailing" testuser
Testing local delivery
.
Cc:
$ sudo -u testuser mail
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/testuser": 1 message 1 new
>N  1 student@serpentar  Sun Oct 30 17:17   14/519   Local mailing
& 1
Message 1:
From student@localhost.localdomain  Sun Oct 30 17:17:46 2011
X-Original-To: testuser
To: testuser@localhost.localdomain
Subject: Local mailing
Date: Sun, 30 Oct 2011 17:17:46 -0400 (EDT)
From: student@localhost.localdomain (student)

Testing local delivery

& quit
```

Проверьте работоспособность виртуальных хостов:

```
$ mail -s "Virtual mailbox - local testing" santa@mail.lan
Test message for virtual user santa
.
Cc:
$ sudo -s
# cat /var/mail/vhosts/mail.lan/santa/new/1320009585.V801I666a7M716203.serpentarium
Return-Path: <student@serpentarium.localdomain>
```

```
X-Original-To: santa@mail.lan
Delivered-To: santa@mail.lan
Received: by serpentarium (Postfix, from userid 1000)
        id 95022666A5; Sun, 30 Oct 2011 17:19:45 -0400 (EDT)
To: santa@mail.lan
Subject: Virtual mailbox - local testing
Message-Id: <20111030211945.95022666A5@serpentarium>
Date: Sun, 30 Oct 2011 17:19:45 -0400 (EDT)
From: student@serpentarium.localdomain (student)

Test message for virtual user santa
```

При необходимости задания квот или smtp аутентификации обратитесь к документации.

3.3.5 Установка и настройка спамфильтра

Добавьте фильтрацию спама с использованием spamassassin. *Необходимо наличие в системе следующих модулей perl: html-parser, net-dns, netaddr-ip, lwp-useragent (perl-HTML-Parser, perl-Net-DNS, perl-NetAddr-IP, perl-LWP-UserAgent-Determined для Fedora, RHEL; libhtml-parser-perl, libnet-dns-perl, libnetaddr-ip-perl, liblwp-useragent-determined-perl для Debian, Ubuntu);* после попадания в консоль perl на все вопросы диалога установки отвечать утвердительно Enter'ом:

```
$ tar xvf Mail-SpamAssassin-<version>.tar.gz
$ cd Mail-SpamAssassin-<version>
$ sudo -s
# perl -MCPAN -e shell
    o conf prerequisites_policy ask
    install Mail::SpamAssassin
    quit
# sa-update
```

Пример конфигурационного файла /etc/mail/spamassassin/local.cf:

```
# This is the right place to customize your installation of SpamAssassin.
#
# See 'perldoc Mail::SpamAssassin::Conf' for details of what can be
# tweaked.
#
# Only a small subset of options are listed below
#
###########################################################################

#   Add *****SPAM***** to the Subject header of spam e-mails
#
rewrite_header Subject *****SPAM*****


#   Save spam messages as a message/rfc822 MIME attachment instead of
#   modifying the original message (0: off, 2: use text/plain instead)
#
report_safe 0


#   Set which networks or hosts are considered 'trusted' by your mail
#   server (i.e. not spammers)
#
# trusted_networks 212.17.35.


#   Set file-locking method (flock is not safe over NFS, but is faster)
#
# lock_method flock


#   Set the threshold at which a message is considered spam (default: 5.0)
#
required_score 1.0
```

```
#   Use Bayesian classifier (default: 1)
#
use_bayes 1


#   Bayesian classifier auto-learning (default: 1)
#
bayes_auto_learn 1


#   Set headers which may provide inappropriate cues to the Bayesian
#   classifier
#
bayes_ignore_header X-Bogosity
bayes_ignore_header X-Spam-Flag
bayes_ignore_header X-Spam-Status


#   Some shortcircuiting, if the plugin is enabled
#
ifplugin Mail::SpamAssassin::Plugin::Shortcircuit
#
#   default: strongly-whitelisted mails are *really* whitelisted now, if the
#   shortcircuiting plugin is active, causing early exit to save CPU load.
#   Uncomment to turn this on
#
# shortcircuit USER_IN_WHITELIST       on
# shortcircuit USER_IN_DEF_WHITELIST   on
# shortcircuit USER_IN_ALL_SPAM_TO     on
# shortcircuit SUBJECT_IN_WHITELIST    on

#   the opposite; blacklisted mails can also save CPU
#
# shortcircuit USER_IN_BLACKLIST       on
# shortcircuit USER_IN_BLACKLIST_TO    on
# shortcircuit SUBJECT_IN_BLACKLIST    on

#   if you have taken the time to correctly specify your "trusted_networks",
#   this is another good way to save CPU
#
# shortcircuit ALL_TRUSTED             on

#   and a well-trained bayes DB can save running rules, too
#
# shortcircuit BAYES_99                spam
# shortcircuit BAYES_00                ham

endif # Mail::SpamAssassin::Plugin::Shortcircuit
```

В данном примере были раскоментированы опции rewrite_header, report_safe, use_bayes, bayes_auto_learn, bayes_ignore_header. Назначение этих опций прокомментировано в конфигурационном файле выше и не требуют пояснений. Значение required_score уменьшено для более строгой фильтрации.

Запустите spamassassin:

```
# spamd -d
```

3.3.6 Интеграция спамфильтра в postfix

Проведите простейшую интеграцию spamassassin в postfix. В данной лабораторной работе спамфильтр будет выполнять постобработку при попадании письма в очередь MTA. Постобработка контента при помощи спамфильтров приведена на рисунке 3.1.

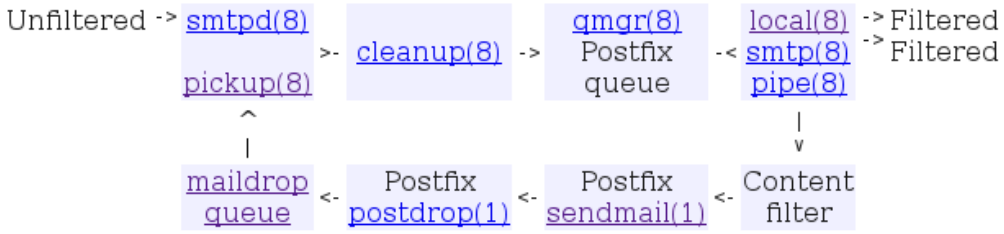

Рисунок 3.1 — Постобработка контента при помощи спамфильтров

С архитектурой postfix можно ознакомится тут: http://www.postfix.org/OVERVIEW.html.

Контентфильтр будет передавать письма из очереди в spamassassin при помощи клиента spamc, результат фильтрации (видоизмененное письмо) отсылать назад в очередь писем postfix при помощи sendmail. Sendmail инжектирует письма таким образом, что повторная передача его спамфильтру будет невозможна.

Отредактируйте файл настроек программных каналов передачи писем.

Пример конфигурационного файла /etc/postfix/master.cf:

```
#
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master").
#
# Do not forget to execute "postfix reload" after editing this file.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
  -o content_filter=spamfilter:dummy
#smtp      inet  n       -       n       -       1       postscreen
#smtpd     pass  -       -       n       -       -       smtpd
#dnsblog   unix  -       -       n       -       0       dnsblog
#tlsproxy  unix  -       -       n       -       0       tlsproxy
#submission inet n       -       n       -       -       smtpd
#  -o smtpd_tls_security_level=encrypt
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#  -o milter_macro_daemon_name=ORIGINATING
#smtps     inet  n       -       n       -       -       smtpd
#  -o smtpd_tls_wrappermode=yes
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#  -o milter_macro_daemon_name=ORIGINATING
#628       inet  n       -       n       -       -       qmqpd
pickup    fifo  n       -       n       60      1       pickup
cleanup   unix  n       -       n       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
#qmgr     fifo  n       -       n       300     1       oqmgr
tlsmgr    unix  -       -       n       1000?   1       tlsmgr
rewrite   unix  -       -       n       -       -       trivial-rewrite
bounce    unix  -       -       n       -       0       bounce
defer     unix  -       -       n       -       0       bounce
trace     unix  -       -       n       -       0       bounce
verify    unix  -       -       n       -       1       verify
flush     unix  n       -       n       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
proxywrite unix -       -       n       -       1       proxymap
smtp      unix  -       -       n       -       -       smtp
# When relaying mail as backup MX, disable fallback_relay to avoid MX loops
relay     unix  -       -       n       -       -       smtp
        -o smtp_fallback_relay=
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix  n       -       n       -       -       showq
error     unix  -       -       n       -       -       error
retry     unix  -       -       n       -       -       error
discard   unix  -       -       n       -       -       discard
```

```
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       n       -       -       lmtp
anvil     unix  -       -       n       -       1       anvil
scache    unix  -       -       n       -       1       scache
#
# ====================================================================
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# Many of the following services use the Postfix pipe(8) delivery
# agent.  See the pipe(8) man page for information about ${recipient}
# and other message envelope options.
# ====================================================================
#
# maildrop. See the Postfix MAILDROP_README file for details.
# Also specify in main.cf: maildrop_destination_recipient_limit=1
#
#maildrop  unix  -       n       n       -       -       pipe
#  flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
#
# ====================================================================
#
# Recent Cyrus versions can use the existing "lmtp" master.cf entry.
#
# Specify in cyrus.conf:
#   lmtp    cmd="lmtpd -a" listen="localhost:lmtp" proto=tcp4
#
# Specify in main.cf one or more of the following:
#  mailbox_transport = lmtp:inet:localhost
#  virtual_transport = lmtp:inet:localhost
#
# ====================================================================
#
# Cyrus 2.1.5 (Amos Gouaux)
# Also specify in main.cf: cyrus_destination_recipient_limit=1
#
#cyrus     unix  -       n       n       -       -       pipe
#  user=cyrus argv=/cyrus/bin/deliver -e -r ${sender} -m ${extension} ${user}
#
# ====================================================================
#
# Old example of delivery via Cyrus.
#
#old-cyrus unix  -       n       n       -       -       pipe
#  flags=R user=cyrus argv=/cyrus/bin/deliver -e -m ${extension} ${user}
#
# ====================================================================
#
# See the Postfix UUCP_README file for configuration details.
#
#uucp      unix  -       n       n       -       -       pipe
#  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail ($recipient)
#
# ====================================================================
#
# Other external delivery methods.
#
#ifmail    unix  -       n       n       -       -       pipe
#  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
#
#bsmtp     unix  -       n       n       -       -       pipe
#  flags=Fq. user=bsmtp argv=/usr/local/sbin/bsmtp -f $sender $nexthop $recipient
#
#scalemail-backend unix -       n       n       -       2       pipe
#  flags=R user=scalemail argv=/usr/lib/scalemail/bin/scalemail-store
#  ${nexthop} ${user} ${extension}
#
#mailman   unix  -       n       n       -       -       pipe
#  flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
#  ${nexthop} ${user}

spamfilter  unix   -       n       n       -       -       pipe
  flags=Rq user=spamfilter argv=/usr/local/bin/spamc -e /usr/sbin/sendmail -i -f
  ${sender} -- ${recipient}
```

Поскольку цепочка фильтрации спама работает от пользователя spamfilter, то создадим его:

```
# useradd -s /bin/false -d /dev/null -g nogroup spamfilter
```

Не забудьте отдать команду postfix, чтобы демон перечитал конфигурацию:

```
# postfix reload
```

3.3.7 *Интеграция антивируса в postfix*

Проведите простейшую интеграцию clamav (clamav-milter) в postfix. В данной лабораторной работе спамфильтр будет выполнять предобработку до попадания письма в очередь MTA. Предобработка контента при помощи мэйл фильтров приведена на рисунке 3.2.

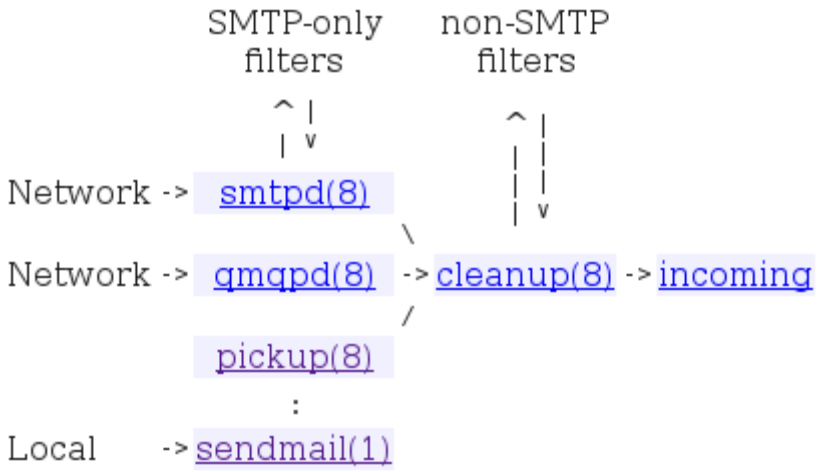

Рисунок 3.2 — Предобработка контента при помощи мэйл фильтров

Необходимо убедиться в наличии development версии библиотеки libmilter (libmilter-dev для Debian, Ubuntu; отсутствует в Fedora, RHEL, что делает сборку невозможной и пользователям данных дистрибутивов следует просто установить бинарный пакет clamav-milter). Пользователя clamav нужно создавать до выполнения сборки. Выполните сборку и установку clamav:

```
$ tar xvf clamav-<version>.tar.gz
$ cd ./clamav-<version>
$ sudo useradd -d /dev/null -s /bin/false -M clamav
$ ./configure --enable-milter —prefix=/ --exec-prefix=/usr —datarootdir=/usr/share
$ make
$ sudo -s
# make install
```

Интеграция антивируса clamav (clamav-milter) проводится предельно просто. Добавьте в конец конфигурационного файла /etc/postfix/main.cf следующие строки:

```
smtpd_milters = unix:/var/run/clamav/clamav-milter.ctl
milter_default_action = accept
```

Примечание 1: большинство демонов postfix выполняются в chroot (/var/spool/postfix). Проверим, в какой среде выполняется smtpd (5й столбик):

```
$ head -n 15 /etc/postfix/master.cf
```

```
#
# Postfix master process configuration file.  For details on the format
# of the file, see the master(5) manual page (command: "man 5 master").
#
# Do not forget to execute "postfix reload" after editing this file.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
  -o content_filter=spamfilter:dummy
#smtp      inet  n       -       n       -       1       postscreen
#smtpd     pass  -       -       n       -       -       smtpd
#dnsblog   unix  -       -       n       -       0       dnsblog
```

Согласно конфигурации smtpd выполняется привилигированным (no chroot), что значит, что поиск сокета clamav-milter в действительности будет происходить по пути /var/run/clamav/clamav-milter.ctl

Сконфигурируйте clamav, clamav-milter, freshclam. Конфигурирование сводится к созданию фалов логов, указания соответствующих путей в конфигурационным файлам, *а также комментированием строк Example.*

```
# mkdir -p /var/lib/clamav
# chown clamav /var/lib/clamav/
# touch /var/log/freshclam.log
# touch /var/log/clamd.log
# touch /var/log/clamav-milter.log
# chown clamav /var/log/*clam*
# mkdir -p /var/run/clamav
# chown clamav /var/run/clamav
```

Пример конфигурационного файла /etc/clamav-milter.conf:

```
##
## Example config file for clamav-milter
##

# Comment or remove the line below.
#Example


##
## Main options
##

# Define the interface through which we communicate with sendmail
# This option is mandatory! Possible formats are:
# [[unix|local]:]/path/to/file - to specify a unix domain socket
# inet:port@[hostname|ip-address] - to specify an ipv4 socket
# inet6:port@[hostname|ip-address] - to specify an ipv6 socket
#
# Default: no default
MilterSocket /var/run/clamav/clamav-milter.ctl
#MilterSocket inet:7357

# Define the group ownership for the (unix) milter socket.
# Default: disabled (the primary group of the user running clamd)
#MilterSocketGroup virusgroup

# Sets the permissions on the (unix) milter socket to the specified mode.
# Default: disabled (obey umask)
MilterSocketMode 666

# Remove stale socket after unclean shutdown.
#
# Default: yes
#FixStaleSocket yes

# Run as another user (clamav-milter must be started by root for this option to work)
#
# Default: unset (don't drop privileges)
User clamav
```

```
        # Initialize supplementary group access (clamav-milter must be started by root).
        #
        # Default: no
        #AllowSupplementaryGroups no

        # Waiting for data from clamd will timeout after this time (seconds).
        # Value of 0 disables the timeout.
        #
        # Default: 120
        #ReadTimeout 300

        # Don't fork into background.
        #
        # Default: no
        #Foreground yes

        # Chroot to the specified directory.
        # Chrooting is performed just after reading the config file and before dropping
privileges.
        #
        # Default: unset (don't chroot)
        #Chroot /newroot

        # This option allows you to save a process identifier of the listening
        # daemon (main thread).
        #
        # Default: disabled
        PidFile /var/run/clamav/clamav-milter.pid

        # Optional path to the global temporary directory.
        # Default: system specific (usually /tmp or /var/tmp).
        #
        #TemporaryDirectory /var/tmp

        ##
        ## Clamd options
        ##

        # Define the clamd socket to connect to for scanning.
        # This option is mandatory! Syntax:
        # ClamdSocket unix:path
        # ClamdSocket tcp:host:port
        # The first syntax specifies a local unix socket (needs an absolute path) e.g.:
        #     ClamdSocket unix:/var/run/clamd/clamd.socket
        # The second syntax specifies a tcp local or remote tcp socket: the
        # host can be a hostname or an ip address; the ":port" field is only required
        # for IPv6 addresses, otherwise it defaults to 3310, e.g.:
        #     ClamdSocket tcp:192.168.0.1
        #
        # This option can be repeated several times with different sockets or even
        # with the same socket: clamd servers will be selected in a round-robin fashion.
        #
        # Default: no default
        ClamdSocket unix:/var/run/clamav/clamd.ctl


        ##
        ## Exclusions
        ##

        # Messages originating from these hosts/networks will not be scanned
        # This option takes a host(name)/mask pair in CIRD notation and can be
        # repeated several times. If "/mask" is omitted, a host is assumed.
        # To specify a locally orignated, non-smtp, email use the keyword "local"
        #
        # Default: unset (scan everything regardless of the origin)
        #LocalNet local
        #LocalNet 192.168.0.0/24
        #LocalNet 1111:2222:3333::/48

        # This option specifies a file which contains a list of basic POSIX regular
        # expressions. Addresses (sent to or from - see below) matching these regexes
        # will not be scanned.  Optionally each line can start with the string "From:"
        # or "To:" (note: no whitespace after the colon) indicating if it is,
        # respectively, the sender or recipient that is to be whitelisted.
        # If the field is missing, "To:" is assumed.
        # Lines starting with #, : or ! are ignored.
        #
        # Default unset (no exclusion applied)
        #Whitelist /etc/whitelisted_addresses
```

```
# Messages from authenticated SMTP users matching this extended POSIX
# regular expression (egrep-like) will not be scanned.
# As an alternative, a file containing a plain (not regex) list of names (one
# per line) can be specified using the prefix "file:".
# e.g. SkipAuthenticated file:/etc/good_guys
#
# Note: this is the AUTH login name!
#
# Default: unset (no whitelisting based on SMTP auth)
#SkipAuthenticated ^(tom|dick|henry)$

# Messages larger than this value won't be scanned.
# Make sure this value is lower or equal than StreamMaxLength in clamd.conf
#
# Default: 25M
#MaxFileSize 10M


##
## Actions
##

# The following group of options controls the delievery process under
# different circumstances.
# The following actions are available:
# - Accept
#   The message is accepted for delievery
# - Reject
#   Immediately refuse delievery (a 5xx error is returned to the peer)
# - Defer
#   Return a temporary failure message (4xx) to the peer
# - Blackhole (not available for OnFail)
#   Like Accept but the message is sent to oblivion
# - Quarantine (not available for OnFail)
#   Like Accept but message is quarantined instead of being delivered
#
# NOTE: In Sendmail the quarantine queue can be examined via mailq -qQ
# For Postfix this causes the message to be placed on hold
#
# Action to be performed on clean messages (mostly useful for testing)
# Default: Accept
#OnClean Accept

# Action to be performed on infected messages
# Default: Quarantine
#OnInfected Quarantine

# Action to be performed on error conditions (this includes failure to
# allocate data structures, no scanners available, network timeouts,
# unknown scanner replies and the like)
# Default: Defer
#OnFail Defer

# This option allows to set a specific rejection reason for infected messages
# and it's therefore only useful together with "OnInfected Reject"
# The string "%v", if present, will be replaced with the virus name.
# Default: MTA specific
#RejectMsg

# If this option is set to "Replace" (or "Yes"), an "X-Virus-Scanned" and an
# "X-Virus-Status" headers will be attached to each processed message, possibly
# replacing existing headers.
# If it is set to Add, the X-Virus headers are added possibly on top of the
# existing ones.
# Note that while "Replace" can potentially break DKIM signatures, "Add" may
# confuse procmail and similar filters.
# Default: no
AddHeader Replace

# When AddHeader is in use, this option allows to arbitrary set the reported
# hostname. This may be desirable in order to avoid leaking internal names.
# If unset the real machine name is used.
# Default: disabled
#ReportHostname my.mail.server.name

# Execute a command (possibly searching PATH) when an infected message is found.
# The following parameters are passed to the invoked program in this order:
# virus name, queue id, sender, destination, subject, message id, message date.
# Note #1: this requires MTA macroes to be available (see LogInfected below)
```

```
# Note #2: the process is invoked in the context of clamav-milter
# Note #3: clamav-milter will wait for the process to exit. Be quick or fork to
# avoid unnecessary delays in email delievery
# Default: disabled
#VirusAction /usr/local/bin/my_infected_message_handler

##
## Logging options
##

# Uncomment this option to enable logging.
# LogFile must be writable for the user running daemon.
# A full path is required.
#
# Default: disabled
LogFile /var/log/clamav-milter.log

# By default the log file is locked for writing - the lock protects against
# running clamav-milter multiple times.
# This option disables log file locking.
#
# Default: no
#LogFileUnlock yes

# Maximum size of the log file.
# Value of 0 disables the limit.
# You may use 'M' or 'm' for megabytes (1M = 1m = 1048576 bytes)
# and 'K' or 'k' for kilobytes (1K = 1k = 1024 bytes). To specify the size
# in bytes just don't use modifiers.
#
# Default: 1M
#LogFileMaxSize 2M

# Log time with each message.
#
# Default: no
LogTime yes

# Use system logger (can work together with LogFile).
#
# Default: no
#LogSyslog yes

# Specify the type of syslog messages - please refer to 'man syslog'
# for facility names.
#
# Default: LOG_LOCAL6
#LogFacility LOG_MAIL

# Enable verbose logging.
#
# Default: no
#LogVerbose yes

# This option allows to tune what is logged when a message is infected.
# Possible values are Off (the default - nothing is logged),
# Basic (minimal info logged), Full (verbose info logged)
# Note:
# For this to work properly in sendmail, make sure the msg_id, mail_addr,
# rcpt_addr and i macroes are available in eom. In other words add a line like:
# Milter.macros.eom={msg_id}, {mail_addr}, {rcpt_addr}, i
# to your .cf file. Alternatively use the macro:
# define(`confMILTER_MACROS_EOM', `{msg_id}, {mail_addr}, {rcpt_addr}, i')
# Postfix should be working fine with the default settings.
#
# Default: disabled
#LogInfected Basic

# This option allows to tune what is logged when no threat is found in a scanned message.
# See LogInfected for possible values and caveats.
# Useful in debugging but drastically increases the log size.
# Default: disabled
#LogClean Basic
```

Пример конфигурационного файла /etc/clamd.conf:

```
##
## Example config file for the Clam AV daemon
```

```
## Please read the clamd.conf(5) manual before editing this file.
##


# Comment or remove the line below.
#Example

# Uncomment this option to enable logging.
# LogFile must be writable for the user running daemon.
# A full path is required.
# Default: disabled
LogFile /var/log/clamd.log

# By default the log file is locked for writing - the lock protects against
# running clamd multiple times (if want to run another clamd, please
# copy the configuration file, change the LogFile variable, and run
# the daemon with --config-file option).
# This option disables log file locking.
# Default: no
#LogFileUnlock yes

# Maximum size of the log file.
# Value of 0 disables the limit.
# You may use 'M' or 'm' for megabytes (1M = 1m = 1048576 bytes)
# and 'K' or 'k' for kilobytes (1K = 1k = 1024 bytes). To specify the size
# in bytes just don't use modifiers.
# Default: 1M
#LogFileMaxSize 2M

# Log time with each message.
# Default: no
LogTime yes

# Also log clean files. Useful in debugging but drastically increases the
# log size.
# Default: no
#LogClean yes

# Use system logger (can work together with LogFile).
# Default: no
#LogSyslog yes

# Specify the type of syslog messages - please refer to 'man syslog'
# for facility names.
# Default: LOG_LOCAL6
#LogFacility LOG_MAIL

# Enable verbose logging.
# Default: no
#LogVerbose yes

# Log additional information about the infected file, such as its
# size and hash, together with the virus name.
#ExtendedDetectionInfo yes

# This option allows you to save a process identifier of the listening
# daemon (main thread).
# Default: disabled
PidFile /var/run/clamav/clamd.pid

# Optional path to the global temporary directory.
# Default: system specific (usually /tmp or /var/tmp).
#TemporaryDirectory /var/tmp

# Path to the database directory.
# Default: hardcoded (depends on installation options)
DatabaseDirectory /var/lib/clamav

# Only load the official signatures published by the ClamAV project.
# Default: no
#OfficialDatabaseOnly no

# The daemon can work in local mode, network mode or both.
# Due to security reasons we recommend the local mode.

# Path to a local socket file the daemon will listen on.
# Default: disabled (must be specified by a user)
LocalSocket /var/run/clamav/clamd.ctl

# Sets the group ownership on the unix socket.
```

```
        # Default: disabled (the primary group of the user running clamd)
        #LocalSocketGroup virusgroup

        # Sets the permissions on the unix socket to the specified mode.
        # Default: disabled (socket is world accessible)
        #LocalSocketMode 660

        # Remove stale socket after unclean shutdown.
        # Default: yes
        #FixStaleSocket yes

        # TCP port address.
        # Default: no
        #TCPSocket 3310

        # TCP address.
        # By default we bind to INADDR_ANY, probably not wise.
        # Enable the following to provide some degree of protection
        # from the outside world.
        # Default: no
        #TCPAddr 127.0.0.1

        # Maximum length the queue of pending connections may grow to.
        # Default: 200
        #MaxConnectionQueueLength 30

        # Clamd uses FTP-like protocol to receive data from remote clients.
        # If you are using clamav-milter to balance load between remote clamd daemons
        # on firewall servers you may need to tune the options below.

        # Close the connection when the data size limit is exceeded.
        # The value should match your MTA's limit for a maximum attachment size.
        # Default: 25M
        #StreamMaxLength 10M

        # Limit port range.
        # Default: 1024
        #StreamMinPort 30000
        # Default: 2048
        #StreamMaxPort 32000

        # Maximum number of threads running at the same time.
        # Default: 10
        #MaxThreads 20

        # Waiting for data from a client socket will timeout after this time (seconds).
        # Default: 120
        #ReadTimeout 300

        # This option specifies the time (in seconds) after which clamd should
        # timeout if a client doesn't provide any initial command after connecting.
        # Default: 5
        #CommandReadTimeout 5

        # This option specifies how long to wait (in miliseconds) if the send buffer is full.
        # Keep this value low to prevent clamd hanging
        #
        # Default: 500
        #SendBufTimeout 200

        # Maximum number of queued items (including those being processed by MaxThreads threads)
        # It is recommended to have this value at least twice MaxThreads if possible.
        # WARNING: you shouldn't increase this too much to avoid running out  of file
descriptors,
        # the following condition should hold:
        # MaxThreads*MaxRecursion + (MaxQueue - MaxThreads) + 6< RLIMIT_NOFILE (usual max is
1024)
        #
        # Default: 100
        #MaxQueue 200

        # Waiting for a new job will timeout after this time (seconds).
        # Default: 30
        #IdleTimeout 60

        # Don't scan files and directories matching regex
        # This directive can be used multiple times
        # Default: scan all
        #ExcludePath ^/proc/
        #ExcludePath ^/sys/
```

```
# Maximum depth directories are scanned at.
# Default: 15
#MaxDirectoryRecursion 20

# Follow directory symlinks.
# Default: no
#FollowDirectorySymlinks yes

# Follow regular file symlinks.
# Default: no
#FollowFileSymlinks yes

# Scan files and directories on other filesystems.
# Default: yes
#CrossFilesystems yes

# Perform a database check.
# Default: 600 (10 min)
#SelfCheck 600

# Execute a command when virus is found. In the command string %v will
# be replaced with the virus name.
# Default: no
#VirusEvent /usr/local/bin/send_sms 123456789 "VIRUS ALERT: %v"

# Run as another user (clamd must be started by root for this option to work)
# Default: don't drop privileges
User clamav

# Initialize supplementary group access (clamd must be started by root).
# Default: no
#AllowSupplementaryGroups no

# Stop daemon when libclamav reports out of memory condition.
#ExitOnOOM yes

# Don't fork into background.
# Default: no
#Foreground yes

# Enable debug messages in libclamav.
# Default: no
#Debug yes

# Do not remove temporary files (for debug purposes).
# Default: no
#LeaveTemporaryFiles yes

# Detect Possibly Unwanted Applications.
# Default: no
#DetectPUA yes

# Exclude a specific PUA category. This directive can be used multiple times.
# See http://www.clamav.net/support/pua for the complete list of PUA
# categories.
# Default: Load all categories (if DetectPUA is activated)
#ExcludePUA NetTool
#ExcludePUA PWTool

# Only include a specific PUA category. This directive can be used multiple
# times.
# Default: Load all categories (if DetectPUA is activated)
#IncludePUA Spy
#IncludePUA Scanner
#IncludePUA RAT

# In some cases (eg. complex malware, exploits in graphic files, and others),
# ClamAV uses special algorithms to provide accurate detection. This option
# controls the algorithmic detection.
# Default: yes
#AlgorithmicDetection yes


##
## Executable files
##

# PE stands for Portable Executable - it's an executable file format used
# in all 32 and 64-bit versions of Windows operating systems. This option allows
```

```
# ClamAV to perform a deeper analysis of executable files and it's also
# required for decompression of popular executable packers such as UPX, FSG,
# and Petite. If you turn off this option, the original files will still be
# scanned, but without additional processing.
# Default: yes
#ScanPE yes

# Executable and Linking Format is a standard format for UN*X executables.
# This option allows you to control the scanning of ELF files.
# If you turn off this option, the original files will still be scanned, but
# without additional processing.
# Default: yes
#ScanELF yes

# With this option clamav will try to detect broken executables (both PE and
# ELF) and mark them as Broken.Executable.
# Default: no
#DetectBrokenExecutables yes


##
## Documents
##

# This option enables scanning of OLE2 files, such as Microsoft Office
# documents and .msi files.
# If you turn off this option, the original files will still be scanned, but
# without additional processing.
# Default: yes
#ScanOLE2 yes


# With this option enabled OLE2 files with VBA macros, which were not
# detected by signatures will be marked as "Heuristics.OLE2.ContainsMacros".
# Default: no
#OLE2BlockMacros no

# This option enables scanning within PDF files.
# If you turn off this option, the original files will still be scanned, but
# without decoding and additional processing.
# Default: yes
#ScanPDF yes


##
## Mail files
##

# Enable internal e-mail scanner.
# If you turn off this option, the original files will still be scanned, but
# without parsing individual messages/attachments.
# Default: yes
#ScanMail yes

# Scan RFC1341 messages split over many emails.
# You will need to periodically clean up $TemporaryDirectory/clamav-partial directory.
# WARNING: This option may open your system to a DoS attack.
#          Never use it on loaded servers.
# Default: no
#ScanPartialMessages yes


# With this option enabled ClamAV will try to detect phishing attempts by using
# signatures.
# Default: yes
#PhishingSignatures yes

# Scan URLs found in mails for phishing attempts using heuristics.
# Default: yes
#PhishingScanURLs yes

# Always block SSL mismatches in URLs, even if the URL isn't in the database.
# This can lead to false positives.
#
# Default: no
#PhishingAlwaysBlockSSLMismatch no

# Always block cloaked URLs, even if URL isn't in database.
# This can lead to false positives.
#
```

```
# Default: no
#PhishingAlwaysBlockCloak no

# Allow heuristic match to take precedence.
# When enabled, if a heuristic scan (such as phishingScan) detects
# a possible virus/phish it will stop scan immediately. Recommended, saves CPU
# scan-time.
# When disabled, virus/phish detected by heuristic scans will be reported only at
# the end of a scan. If an archive contains both a heuristically detected
# virus/phish, and a real malware, the real malware will be reported
#
# Keep this disabled if you intend to handle "*.Heuristics.*" viruses
# differently from "real" malware.
# If a non-heuristically-detected virus (signature-based) is found first,
# the scan is interrupted immediately, regardless of this config option.
#
# Default: no
#HeuristicScanPrecedence yes

##
## Data Loss Prevention (DLP)
##

# Enable the DLP module
# Default: No
#StructuredDataDetection yes

# This option sets the lowest number of Credit Card numbers found in a file
# to generate a detect.
# Default: 3
#StructuredMinCreditCardCount 5

# This option sets the lowest number of Social Security Numbers found
# in a file to generate a detect.
# Default: 3
#StructuredMinSSNCount 5

# With this option enabled the DLP module will search for valid
# SSNs formatted as xxx-yy-zzzz
# Default: yes
#StructuredSSNFormatNormal yes

# With this option enabled the DLP module will search for valid
# SSNs formatted as xxxyyzzzz
# Default: no
#StructuredSSNFormatStripped yes


##
## HTML
##

# Perform HTML normalisation and decryption of MS Script Encoder code.
# Default: yes
# If you turn off this option, the original files will still be scanned, but
# without additional processing.
#ScanHTML yes


##
## Archives
##

# ClamAV can scan within archives and compressed files.
# If you turn off this option, the original files will still be scanned, but
# without unpacking and additional processing.
# Default: yes
#ScanArchive yes

# Mark encrypted archives as viruses (Encrypted.Zip, Encrypted.RAR).
# Default: no
#ArchiveBlockEncrypted no


##
## Limits
##

# The options below protect your system against Denial of Service attacks
# using archive bombs.
```

```
# This option sets the maximum amount of data to be scanned for each input file.
# Archives and other containers are recursively extracted and scanned up to this
# value.
# Value of 0 disables the limit
# Note: disabling this limit or setting it too high may result in severe damage
# to the system.
# Default: 100M
#MaxScanSize 150M

# Files larger than this limit won't be scanned. Affects the input file itself
# as well as files contained inside it (when the input file is an archive, a
# document or some other kind of container).
# Value of 0 disables the limit.
# Note: disabling this limit or setting it too high may result in severe damage
# to the system.
# Default: 25M
#MaxFileSize 30M

# Nested archives are scanned recursively, e.g. if a Zip archive contains a RAR
# file, all files within it will also be scanned. This options specifies how
# deeply the process should be continued.
# Note: setting this limit too high may result in severe damage to the system.
# Default: 16
#MaxRecursion 10

# Number of files to be scanned within an archive, a document, or any other
# container file.
# Value of 0 disables the limit.
# Note: disabling this limit or setting it too high may result in severe damage
# to the system.
# Default: 10000
#MaxFiles 15000


##
## Clamuko settings
##

# Enable Clamuko. Dazuko must be configured and running. Clamuko supports
# both Dazuko (/dev/dazuko) and DazukoFS (/dev/dazukofs.ctrl). DazukoFS
# is the preferred option. For more information please visit www.dazuko.org
# Default: no
#ClamukoScanOnAccess yes

# The number of scanner threads that will be started (DazukoFS only).
# Having multiple scanner threads allows Clamuko to serve multiple
# processes simultaneously. This is particularly beneficial on SMP machines.
# Default: 3
#ClamukoScannerCount 3

# Don't scan files larger than ClamukoMaxFileSize
# Value of 0 disables the limit.
# Default: 5M
#ClamukoMaxFileSize 10M

# Set access mask for Clamuko (Dazuko only).
# Default: no
#ClamukoScanOnOpen yes
#ClamukoScanOnClose yes
#ClamukoScanOnExec yes

# Set the include paths (all files inside them will be scanned). You can have
# multiple ClamukoIncludePath directives but each directory must be added
# in a seperate line. (Dazuko only)
# Default: disabled
#ClamukoIncludePath /home
#ClamukoIncludePath /students

# Set the exclude paths. All subdirectories are also excluded. (Dazuko only)
# Default: disabled
#ClamukoExcludePath /home/bofh

# With this option you can whitelist specific UIDs. Processes with these UIDs
# will be able to access all files.
# This option can be used multiple times (one per line).
# Default: disabled
#ClamukoExcludeUID 0

# With this option enabled ClamAV will load bytecode from the database.
```

```
        # It is highly recommended you keep this option on, otherwise you'll miss detections for
many new viruses.
        # Default: yes
        #Bytecode yes

        # Set bytecode security level.
        # Possible values:
        #       None - no security at all, meant for debugging. DO NOT USE THIS ON PRODUCTION
SYSTEMS
        #         This value is only available if clamav was built with --enable-debug!
        #       TrustSigned - trust bytecode loaded from signed .c[lv]d files,
        #                insert runtime safety checks for bytecode loaded from other sources
        #       Paranoid - don't trust any bytecode, insert runtime checks for all
        # Recommended: TrustSigned, because bytecode in .cvd files already has these checks
        # Note that by default only signed bytecode is loaded, currently you can only
        # load unsigned bytecode in --enable-debug mode.
        #
        # Default: TrustSigned
        #BytecodeSecurity TrustSigned

        # Set bytecode timeout in miliseconds.
        #
        # Default: 5000
        # BytecodeTimeout 1000
```

Пример конфигурационного файла /etc/freshclam.conf:

```
        ##
        ## Example config file for freshclam
        ## Please read the freshclam.conf(5) manual before editing this file.
        ##


        # Comment or remove the line below.
        #Example

        # Path to the database directory.
        # WARNING: It must match clamd.conf's directive!
        # Default: hardcoded (depends on installation options)
        DatabaseDirectory /var/lib/clamav

        # Path to the log file (make sure it has proper permissions)
        # Default: disabled
        UpdateLogFile /var/log/freshclam.log

        # Maximum size of the log file.
        # Value of 0 disables the limit.
        # You may use 'M' or 'm' for megabytes (1M = 1m = 1048576 bytes)
        # and 'K' or 'k' for kilobytes (1K = 1k = 1024 bytes).
        # in bytes just don't use modifiers.
        # Default: 1M
        #LogFileMaxSize 2M

        # Log time with each message.
        # Default: no
        LogTime yes

        # Enable verbose logging.
        # Default: no
        #LogVerbose yes

        # Use system logger (can work together with UpdateLogFile).
        # Default: no
        #LogSyslog yes

        # Specify the type of syslog messages - please refer to 'man syslog'
        # for facility names.
        # Default: LOG_LOCAL6
        #LogFacility LOG_MAIL

        # This option allows you to save the process identifier of the daemon
        # Default: disabled
        PidFile /var/run/clamav/freshclam.pid

        # By default when started freshclam drops privileges and switches to the
        # "clamav" user. This directive allows you to change the database owner.
        # Default: clamav (may depend on installation options)
        DatabaseOwner clamav
```

```
# Initialize supplementary group access (freshclam must be started by root).
# Default: no
#AllowSupplementaryGroups yes

# Use DNS to verify virus database version. Freshclam uses DNS TXT records
# to verify database and software versions. With this directive you can change
# the database verification domain.
# WARNING: Do not touch it unless you're configuring freshclam to use your
# own database verification domain.
# Default: current.cvd.clamav.net
#DNSDatabaseInfo current.cvd.clamav.net

# Uncomment the following line and replace XY with your country
# code. See http://www.iana.org/cctld/cctld-whois.htm for the full list.
# You can use db.XY.ipv6.clamav.net for IPv6 connections.
#DatabaseMirror db.XY.clamav.net

# database.clamav.net is a round-robin record which points to our most
# reliable mirrors. It's used as a fall back in case db.XY.clamav.net is
# not working. DO NOT TOUCH the following line unless you know what you
# are doing.
DatabaseMirror database.clamav.net

# How many attempts to make before giving up.
# Default: 3 (per mirror)
#MaxAttempts 5

# With this option you can control scripted updates. It's highly recommended
# to keep it enabled.
# Default: yes
#ScriptedUpdates yes

# By default freshclam will keep the local databases (.cld) uncompressed to
# make their handling faster. With this option you can enable the compression;
# the change will take effect with the next database update.
# Default: no
#CompressLocalDatabase no

# With this option you can provide custom sources (http:// or file://) for
# database files. This option can be used multiple times.
# Default: no custom URLs
#DatabaseCustomURL http://myserver.com/mysigs.ndb
#DatabaseCustomURL file:///mnt/nfs/local.hdb

# Number of database checks per day.
# Default: 12 (every two hours)
#Checks 24

# Proxy settings
# Default: disabled
#HTTPProxyServer myproxy.com
#HTTPProxyPort 1234
#HTTPProxyUsername myusername
#HTTPProxyPassword mypass

# If your servers are behind a firewall/proxy which applies User-Agent
# filtering you can use this option to force the use of a different
# User-Agent header.
# Default: clamav/version_number
#HTTPUserAgent SomeUserAgentIdString

# Use aaa.bbb.ccc.ddd as client address for downloading databases. Useful for
# multi-homed systems.
# Default: Use OS'es default outgoing IP address.
#LocalIPAddress aaa.bbb.ccc.ddd

# Send the RELOAD command to clamd.
# Default: no
NotifyClamd /etc/clamd.conf

# Run command after successful database update.
# Default: disabled
#OnUpdateExecute command

# Run command when database update process fails.
# Default: disabled
#OnErrorExecute command

# Run command when freshclam reports outdated version.
# In the command string %v will be replaced by the new version number.
```

```
# Default: disabled
#OnOutdatedExecute command

# Don't fork into background.
# Default: no
#Foreground yes

# Enable debug messages in libclamav.
# Default: no
#Debug yes

# Timeout in seconds when connecting to database server.
# Default: 30
#ConnectTimeout 60

# Timeout in seconds when reading from database server.
# Default: 30
#ReceiveTimeout 60

# With this option enabled, freshclam will attempt to load new
# databases into memory to make sure they are properly handled
# by libclamav before replacing the old ones.
# Default: yes
#TestDatabases yes

# When enabled freshclam will submit statistics to the ClamAV Project about
# the latest virus detections in your environment. The ClamAV maintainers
# will then use this data to determine what types of malware are the most
# detected in the field and in what geographic area they are.
# Freshclam will connect to clamd in order to get recent statistics.
# Default: no
#SubmitDetectionStats /path/to/clamd.conf

# Country of origin of malware/detection statistics (for statistical
# purposes only). The statistics collector at ClamAV.net will look up
# your IP address to determine the geographical origin of the malware
# reported by your installation. If this installation is mainly used to
# scan data which comes from a different location, please enable this
# option and enter a two-letter code (see http://www.iana.org/domains/root/db/)
# of the country of origin.
# Default: disabled
#DetectionStatsCountry country-code

# This option enables support for our "Personal Statistics" service.
# When this option is enabled, the information on malware detected by
# your clamd installation is made available to you through our website.
# To get your HostID, log on http://www.stats.clamav.net and add a new
# host to your host list. Once you have the HostID, uncomment this option
# and paste the HostID here. As soon as your freshclam starts submitting
# information to our stats collecting service, you will be able to view
# the statistics of this clamd installation by logging into
# http://www.stats.clamav.net with the same credentials you used to
# generate the HostID. For more information refer to:
# http://www.clamav.net/support/faq/faq-cctts/
# This feature requires SubmitDetectionStats to be enabled.
# Default: disabled
#DetectionStatsHostID unique-id

# This option enables support for Google Safe Browsing. When activated for
# the first time, freshclam will download a new database file (safebrowsing.cvd)
# which will be automatically loaded by clamd and clamscan during the next
# reload, provided that the heuristic phishing detection is turned on. This
# database includes information about websites that may be phishing sites or
# possible sources of malware. When using this option, it's mandatory to run
# freshclam at least every 30 minutes.
# Freshclam uses the ClamAV's mirror infrastructure to distribute the
# database and its updates but all the contents are provided under Google's
# terms of use. See http://code.google.com/support/bin/answer.py?answer=70015
# and http://safebrowsing.clamav.net for more information.
# Default: disabled
#SafeBrowsing yes

# This option enables downloading of bytecode.cvd, which includes additional
# detection mechanisms and improvements to the ClamAV engine.
# Default: enabled
#Bytecode yes

# Download an additional 3rd party signature database distributed through
# the ClamAV mirrors. Here you can find a list of available databases:
# http://www.clamav.net/download/cvd/3rdparty
```

```
# This option can be used multiple times.
#ExtraDatabase dbname1
#ExtraDatabase dbname2
```

Запустите clamav, clamav-milter, freshclam (первый запуск freshclam используется для обновления базы сигнатур):

```
# freshclam -v
# clamd
# freshclam -d
# clamav-milter
```

После всего проделанного письма должны фильтроваться от спама и вирусов. Укажите postfix команду для перечитывания конфигурации:

```
# postfix reload
```

Проверьте, все ли в порядке отсылкой очередного письма одному из, например, виртуальных пользователей.

Примечание 2: при использовании клиентов mail, mutt т.д. postfix доставляет письмо в обход smtpd (клиент не соединяется на 25 порт), по этому необходимо отправить письмо «извне» (подключится к нашему smtp по сети). Используйте telnet:

```
$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 localhost ESMTP Postfix
helo localhost
250 localhost
mail from: dummy@mailbox.com
250 2.1.0 Ok
rcpt to: test@mail.lan
250 2.1.5 Ok
data
354 End data with <CR><LF>.<CR><LF>
This message will be passed through spamfilter and clamav
Some new headers will be added
.
250 2.0.0 Ok: queued as D660866712
quit
221 2.0.0 Bye
Connection closed by foreign host.
# sudo -s
# cat /var/mail/vhosts/mail.lan/test/new/1320064241.V801I68036M405968.serpentarium
Return-Path: <dummy@mailbox.com>
X-Original-To: test@mail.lan
Delivered-To: test@mail.lan
Received: by localhost (Postfix, from userid 5003)
        id 480A766715; Mon, 31 Oct 2011 08:30:41 -0400 (EDT)
X-Spam-Checker-Version: SpamAssassin 3.3.2 (2011-06-06) on serpentarium
X-Spam-Flag: YES
X-Spam-Level: *
X-Spam-Status: Yes, score=2.0 required=1.0 tests=ALL_TRUSTED,MISSING_HEADERS,
        MISSING_SUBJECT autolearn=no version=3.3.2
X-Spam-Report:
        * -1.0 ALL_TRUSTED Passed through trusted hosts only via SMTP
        *  1.2 MISSING_HEADERS Missing To: header
        *  1.8 MISSING_SUBJECT Missing Subject: header
Received: from localhost (localhost [127.0.0.1])
        by localhost (Postfix) with SMTP id D660866712
        for <test@mail.lan>; Mon, 31 Oct 2011 08:29:42 -0400 (EDT)
Message-Id: <20111031122953.D660866712@localhost>
Date: Mon, 31 Oct 2011 08:29:42 -0400 (EDT)
From: dummy@mailbox.com
X-Virus-Scanned: clamav-milter 0.97.3 at serpentarium
X-Virus-Status: Clean
Subject: *****SPAM*****
X-Spam-Prev-Subject: (nonexistent)
```

```
This message will be passed through spamfilter and clamav
Some new headers will be added
```

Как видно в полученном письме появились заголовки X-Spam-*, свидетельствующие о просмотре письма spamassassin, и X-Virus-*, что свидетельствует об сканировании письма антивирусом.

3.4 Содержание отчета

Отчёт должен содержать ход выполнения лабораторной работы с листингами лог-файлов серверов Postfix и Clamav-Milter, а также тела оригинальных и профильтрованных писем.

3.5 Контрольные вопросы

1. По какому протоколу сетевого уровня сервер Postfix подключается к серверу FreeRADIUS.
2. Опишите механизмы проверки писем антивирусом и спамфильтром.
3. Возможна ли работа всех демонов Postfix в chroot-окружении?
4. Опишите процесс конфигурирования Postfix для работы в chroot-окружении.
5. Какими методами (кроме chroot-окружения) можно повысить защищенность сервера от атак? Опишите возможности сервера Postfix для защиты клиент-серверных соединений.
6. Назовите причины использования автоизменения адресов? Приведите примеры использования автоизменения адресов в сервере Postfix.
7. Опишите основные подходы для запуска нескольких копий сервера Postfix с разными конфигурациями.
8. Опишите базовую архитектуру сервера Postfix. Опишите процесс получения почты сервером Postfix.
9. Опишите базовую архитектуру сервера Postfix. Опишите процесс доставки почты сервером Postfix.
10. Опишите базовую архитектуру сервера Postfix. Перечислите и кратко опишите основные вспомогательные сервисы сервера Postfix.

4 Самостоятельная работа №4. Преобразование сетевых адресов и статистика в фильтрах iptables. Типовые решения защиты сетей

4.1 Цель работы

Освоить синтаксис команд управления фильтром пакетов в ОС Linux. Промоделировать работу сетевого шлюза между двумя сетями. Освоить технику работы с трансляцией сетевых адресов (NAT). Рассмотреть конфигурации фильтров для стандартных типов подключения сетей.

4.2 Краткие теоретические сведения

Iptables — утилита командной строки, является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) netfilter для ядер Linux версий 2.4, 2.6, 3.0. Для использования утилиты iptables требуются привилегии суперпользователя (root). Иногда под словом iptables имеется в виду и сам межсетевой экран netfilter.

Чтобы добавить новые правила нужно использовать команду iptables-save, которая сохраняет текущие правила в файл конфигурации (*для Fedora и RHEL /etc/sysconfig/iptables используется по-умолчанию, и автоматически подгружается при запуске сервиса iptables*).

Для Debian и Ubuntu, при необходимости, можно вручную настроить конфигурацию, например тут: /etc/iptables.up.rules; а затем настроить автоматическое использование этого файла при запуске системы.

Файл /etc/network/if-pre-up.d/iptables (не забудьте установить execution бит):

```
#!/bin/bash
/sbin/iptables-restore < /etc/iptables.up.rules
```

Пример файла конфигурации iptables:

```
*filter

# Allows all loopback (lo0) traffic and drop all traffic to 127/8 that doesn't use lo0
-A INPUT -i lo -j ACCEPT
-A INPUT ! -i lo -d 127.0.0.0/8 -j REJECT

# Accepts all established inbound connections
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# Allows all outbound traffic
# You could modify this to only allow certain traffic
-A OUTPUT -j ACCEPT

# Allows HTTP and HTTPS connections from anywhere (the normal ports for websites)
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT

# Allows SSH connections for script kiddies
# THE -dport NUMBER IS THE SAME ONE YOU SET UP IN THE SSHD_CONFIG FILE
-A INPUT -p tcp -m state --state NEW --dport 30000 -j ACCEPT

# Now you should read up on iptables rules and consider whether ssh access
# for everyone is really desired. Most likely you will only allow access from certain
IPs.
```

```
# Allow ping
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

# log iptables denied calls (access via 'dmesg' command)
-A INPUT -m limit --limit 5/min -j LOG --log-prefix "iptables denied: " --log-level 7

# Reject all other inbound - default deny unless explicitly allowed policy:
-A INPUT -j REJECT
-A FORWARD -j REJECT

COMMIT
```

Для активации правил из файла конфигурации (не сохраняются после перезапуска системы), необходимо выполнить команду:

```
# iptables-restore < /etc/iptables.test.rules
```

Чтобы вывести список активированных правил нужно выполнить команду iptables с ключом L:

```
# iptables -L -n
```

Чтобы вывести дополнительную информацию о цепочках и правилах, выполните

```
# iptables -L -n -v
```

В операционной системе Linux преобразование сетевых адресов выполняется с помощью iptables в ядре в таблице nat.

Только первый пакет из потока проходит через цепочки этой таблицы, трансляция адресов или маскировка применяются ко всем последующим пакетам в потоке автоматически. Для этой таблицы характерны действия: DNAT, SNAT, MASQUERADE.

Действие DNAT (Destination Network Address Translation) производит преобразование адресов назначения в заголовках пакетов. Этим действием производится перенаправление пакетов на другие адреса, отличные от указанных в заголовках пакетов. Типичным применением является перенаправление http запросов на прокси сервер или перенаправление запросов к сервису по публичному адресу на сервис, расположенный в приватной сети.

SNAT (Source Network Address Translation) используется для изменения исходных адресов пакетов. С помощью этого действия можно скрыть структуру локальной сети, а заодно и разделить единственный внешний IP адрес между компьютерами локальной сети для выхода в Интернет. В этом случае брандмауэр с помощью SNAT автоматически производит прямое и обратное преобразование адресов, тем самым давая возможность выполнять подключение к серверам в Интернете с компьютеров в локальной сети.

Маскировка (MASQUERADE) применяется в тех же целях, что и SNAT, но в отличие от последней, MASQUERADE дает более сильную нагрузку на систему. Происходит это потому что каждый раз, когда требуется выполнение этого действия, производится запрос IP адреса для указанного в действии сетевого интерфейса, в то время как для SNAT IP адрес указывается непосредственно. Однако, благодаря такому отличию,

MASQUERADE может работать в случаях с динамическим IP адресом, т.е. когда вы подключаетесь к Интернет, скажем, через PPP, SLIP или DHCP.

4.3 Ход работы

Вполне естественно, что адреса и некоторые другие аргументы команд в вашем случае будут другие. Не нужно перенабирать команды приведенные далее, необходимо разобраться для чего они предназначены и правильно использовать их для своих параметров сети.

Рекомендуется в качестве Host1 на рисунке 4.1 использовать хост на виртуальной машине к которому в настройках ВМ подключить 2 виртуальных сетевых адаптера в режиме моста.

Обратите внимание, что в VirtualBox может присутствовать ошибка, которая заключается в том, что выданный для одного из интерфейсов ВМ IP адрес обозначается роутером, как дубликат и пропадает подключение ко внешней сети. В случае возникновения данной проблемы необходимо вручную переназначить IP адрес проблемного интерфейса и исправить маршрутизацию.

4.3.1 Установка необходимых пакетов

В большинстве дистрибутивов все необходимые пакеты для выполнения данной работы включены в базовую поставку, если же у вас другая ситуация то проверьте наличие пакетов (iptables, openssh-server, openssh-client для Ubuntu и Debian, iptables, openssh-server, openssh-clients для RHEL и Fedora).

1. Удалите все существующие правила:

```
# iptables -F
# iptables -X
# iptables -t nat -F
# iptables -t nat -X
# iptables -t mangle -F
# iptables -t mangle –X
```

2. Задайте политики по умолчанию для цепочек INPUT, OUTPUT, FORWARD таблицы filter:

```
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT ACCEPT
```

Теперь Ваш компьютер не может принимать входящие пакеты, хотя все исходящие пакеты разрешены. Сетевые программы и сервисы не работают. Проверить это можно следующими командами:

```
$ ping kid.stu.cn.ua
$ ssh <ваш_логин>@kid.stu.cn.ua
```

Ответа не будет.

3. Разрешите входящие пакеты для локальной петли и установленных соединений. Задайте правила:

```
# iptables -A INPUT -i lo -j ACCEPT
# iptables -A INPUT -i eth0 -m conntrack --ctstate RELATED,ESTABLISHED
-j ACCEPT
```

4. Попробуйте выполнить команду:

```
$ ssh <ваш_логин>@kid.stu.cn.ua
```

5. Разрешите команду тестирования сети ping:

```
# iptables -A INPUT -i eth0 -p ICMP -s 0/0 --icmp-type 0 -j ACCEPT
# iptables -A INPUT -i eth0 -p ICMP -s 0/0 --icmp-type 3 -j ACCEPT
# iptables -A INPUT -i eth0 -p ICMP -s 0/0 --icmp-type 5 -j ACCEPT
# iptables -A INPUT -i eth0 -p ICMP -s 0/0 --icmp-type 8 -j ACCEPT
# iptables -A INPUT -i eth0 -p ICMP -s 0/0 --icmp-type 11 -j ACCEPT
```

Действие этих правил можно проверить командой ping *на свой хост*.

6. Разрешите входящие ssh соединения:

```
# iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT
```

Проверить это правило можно запустив на компьютере sshd и зайдя на свой хост удаленно по ssh.

4.3.2 Преобразование сетевых адресов

Необходимо наличие сетевых двух интерфейсов с разными подсетями подключенными к ним. Пример схемы сети приведен на рисунке 4.1.

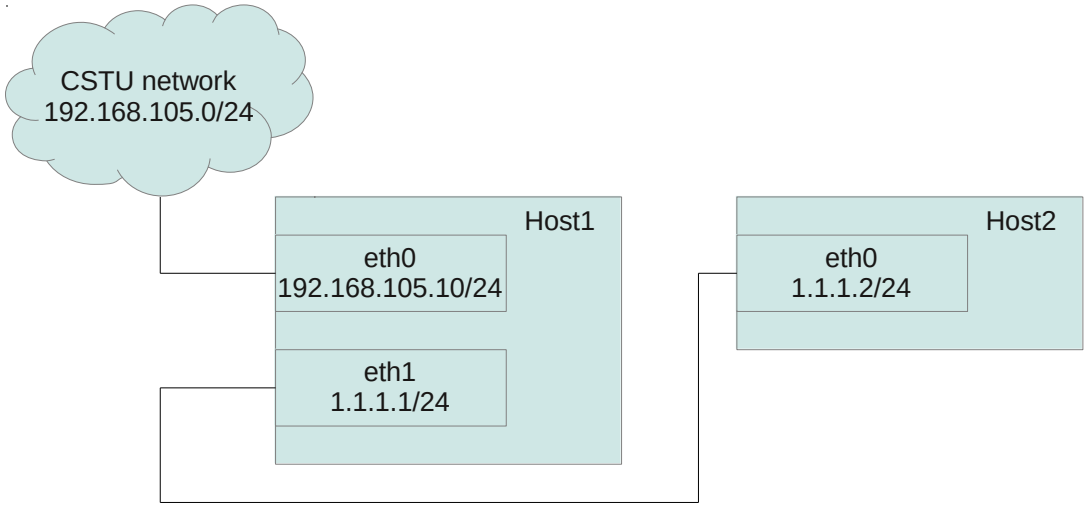

Рисунок 4.1 — Пример схемы сети

Чтобы адаптеры на Host1 могли обмениваться пакетами необходимо включить packet forwarding. Для этого в файле /etc/sysctl.conf значение параметра net.ipv4.ip_forward измените на 1 и перечитайте конфигурацию:

```
# sysctl -p
```

4.3.3 Преобразование адреса при помощи SNAT

Для примера, проиллюстрированного на рисунке 4.1, на Host1 добавьте следующее правило:

```
# iptables -A INPUT -i eth1 -j ACCEPT
```

```
# iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
# iptables -A FORWARD -i eth0 -o eth1 -j ACCEPT
# iptables -t nat -A POSTROUTING -j SNAT --to-source 192.168.105.10
```

Для того, что бы проверить данное правило на Host2 добавьте маршрут по умолчанию через 1.1.1.1 и запустите ping на адрес из внешней сети.

```
# route add default gw 1.1.1.1
```

4.3.4 Преобразование адреса назначения пакета (Destination Network Adress Translation)

Для примера, проиллюстрированного на рисунке 4.1 на Host1 добавьте следующие правила:

```
# iptables -A INPUT -i eth0 -p tcp -d 192.168.105.10 --dport 222 -j
ACCEPT
# iptables -t nat -A PREROUTING -p tcp -d 192.168.105.10 --dport 222 -j
DNAT --to-destination 1.1.1.2:22
```

Для того, чтобы проверить работоспособность данного правила, нужно на Host2 запустить sshd и попробовать выполнить вход:

```
$ ssh <user_from_host_1>@192.168.105.10:222
```

4.3.5 Преобразование адреса при помощи MASQUERADE

Примечание: перед выполнением данного пункта необходимо предварительно удалить правило, добавленное в пункте по настройке SNAT. Сначала выведите список правил из таблицы nat:

```
# iptables -t nat -L --line-numbers
```

Затем удалите все правила из цепочки POSTROUTING:

```
# iptables -t nat -D POSTROUTING <номер_правила_начиная_с_1>
```

Для примера, проиллюстрированного на рисунке 4.1, на Host1 добавьте следующее правило:

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Проверьте работу сети.

4.3.6 Реализация подсчета трафика с помощью iptables на локальном компьютере

Нужно создать две цепочки для подсчета входящего и исходящего трафика соответственно.

```
# iptables -N incount
# iptables -N outcount
```

Добавить правила для переброски трафика в цепочки incount и outcount.

```
# iptables -I INPUT 1 -i eth0 -j incount
# iptables -I OUTPUT 1 -o eth0 -j outcount
```

Правила, которые считают весь входящий и исходящий трафик:

```
# iptables -A incount -i eth0 -s 0/0 -d 0/0
# iptables -A outcount -o eth0 -s 0/0 -d 0/0
```

Просмотреть статистику по правилам цепочки можно командами:

```
# iptables -L outcount -v -x --line-numbers -n
# iptables -L incount -v -x --line-number -n
```

Обнулить счетчики правил в цепочке можно командой:

```
# iptables -L outcount -Z
```

4.3.7 Конфигурирование типового сетевого экрана для домашней локальной сети с выходом в интернет через PPTP подключение

Устно проанализируйте приведенные ниже варианты конфигураций для Iptables, выполнять приведенные правила не нужно. Приведите в отчете варианты применения для первого и второго варианта конфигурации.

Вариант 1:

```
#!/bin/sh
#
# rc.firewall - Initial SIMPLE IP Firewall script

IPTABLES="/sbin/iptables"

$IPTABLES -P INPUT DROP
$IPTABLES -P FORWARD DROP
$IPTABLES -P OUTPUT ACCEPT

$IPTABLES -t nat POSTROUTING -o ppp0 -j MASQUERADE

$IPTABLES -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A INPUT -p ALL -i lo -j ACCEPT

$IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -p ALL -o eth0 -j ACCEPT
```

Вариант 2:

```
#!/bin/sh
#
# rc.firewall - Initial SIMPLE IP Firewall script
#
# Authors: Polianitsa Yuriy, Lavrenenko Pavlo
# Based on David Whitmarsh script
# (c) 2001, 2002 Sparkle Computer Co ltd.
# based on rc.firewall by Oskar Andreasson <blueflux@koffein.net>

############
# Секция задания нужной конфигурации  Вашего компьютера ускорит запуск
# приведенного сценария фильтра.

#
# Диапазон значений Вашей локальной сети и localhost IP. /24 означает,
# что под адрес сети используются только первых 24 бита 32-го  IP
#адреса,что эквивалентно 255.255.255.0

LAN_BCAST_ADDRESS="xxx.xxx.xxx.63"
INTERNAL_ADDRESS_RANGE="xxx.xxx.xxx.56/29"

INET_IFACE="eth1"
LAN_IFACE="eth0"

LO_IFACE="lo"
LO_IP="127.0.0.1"
```

```
        IPTABLES="/sbin/iptables"

        # Очистить iptables от правил

        $IPTABLES -F
        $IPTABLES -X
        $IPTABLES -t nat -F
        $IPTABLES -t nat -X
        $IPTABLES -t mangle -F
        $IPTABLES -t mangle -X

        #
        # Установить политики по-умолчанию для цепочек  INPUT, FORWARD и OUTPUT
        #

        $IPTABLES -P INPUT DROP
        $IPTABLES -P OUTPUT ACCEPT
        $IPTABLES -P FORWARD DROP

        # Начнем с таблицы mangle
        # цепочка PREROUTING

        $IPTABLES -t mangle -P PREROUTING DROP

        # Запретить IP spoofing

        $IPTABLES -t mangle -A PREROUTING -s 192.168.0.0/16 -j DROP
        $IPTABLES -t mangle -A PREROUTING -s 10.0.0.0/8 -j DROP
        $IPTABLES -t mangle -A PREROUTING -s 172.16.0.0/12 -j DROP

        # Разрешить внутренние пакеты на внутреннем сетевом интерфейсе
        $IPTABLES -t mangle -A PREROUTING -i $LAN_IFACE -s $INTERNAL_ADDRESS_RANGE -j ACCEPT

        # Разрешить внешние пакеты на веншнем интерфейсе
        $IPTABLES -t mangle -A PREROUTING -i $INET_IFACE ! -s $INTERNAL_ADDRESS_RANGE -j ACCEPT

        # Разрешить транзит пакетов
        $IPTABLES -A FORWARD -p ALL -s $INTERNAL_ADDRESS_RANGE -j ACCEPT
        $IPTABLES -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
        $IPTABLES -A FORWARD -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level 7 --
log-prefix "IPT FORWARD packet died: "


        # Создать специальные цепочки для  ICMP, TCP и UDP пакетов

        $IPTABLES -N icmp_packets
        #
        # правила для ICMP пакетов
        #

        # Разрешить тип ICMP пакетов echo reply
        $IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 0 -j ACCEPT
        # Разрешить тип ICMP пакетов destination  unreachable
        $IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 3 -j ACCEPT
        # Разрешить тип ICMP пакетов redirect
        $IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 5 -j ACCEPT
        # Разрешить тип ICMP пакетов time exceeded
        $IPTABLES -A icmp_packets -p ICMP -s 0/0 --icmp-type 11 -j ACCEPT
        $IPTABLES -A FORWARD -p ICMP -j icmp_packets

        #
        # правила для UDP пакетов
        #
        $IPTABLES -N udpincoming_packets

        $IPTABLES -A udpincoming_packets -p UDP -s 0/0 --source-port 53 -j ACCEPT       # DNS
        $IPTABLES -A udpincoming_packets -p UDP -s 0/0 --source-port 123 -j ACCEPT      # ntp

        $IPTABLES -A FORWARD -p UDP -j udpincoming_packets

        # правила для TCP пакетов

        $IPTABLES -N tcp_packets

        #
        # Цепочка  allowed для разрешенных TCP соединений
        #

        $IPTABLES -N allowed
```

```
        $IPTABLES -A allowed -p TCP --syn -j ACCEPT
        $IPTABLES -A allowed -p TCP -m state --state ESTABLISHED,RELATED -j ACCEPT
        $IPTABLES -A allowed -p TCP -j DROP

        # Журналировать и отбросить  TCP пакеты с признаком New и сброшеннм
        # битом SYN
        $IPTABLES -A tcp_packets -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "New not
syn:"
        $IPTABLES -A tcp_packets -p tcp ! --syn -m state --state NEW -j DROP

        $IPTABLES -A FORWARD -p TCP -j tcp_packets

        # Обязательное правило для интерфейса петля
        $IPTABLES -A INPUT -p ALL -i $LO_IFACE -d $LO_IP -j ACCEPT

        $IPTABLES -A INPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level 7 --log-
prefix "IPT INPUT packet died: "

        #
        # цепочка OUTPUT
        #

        $IPTABLES -A OUTPUT -p tcp ! --syn -m state --state NEW -j LOG --log-prefix "New not
syn:"
        $IPTABLES -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

        $IPTABLES -A OUTPUT -p ALL -s $LO_IP -j ACCEPT
        $IPTABLES -A OUTPUT -p ALL -s $BR_IP -j ACCEPT
        $IPTABLES -A OUTPUT -m limit --limit 3/minute --limit-burst 3 -j LOG --log-level 7 --log-
prefix "IPT OUTPUT packet died: "
```

4.4 Содержание отчета

Отчёт должен содержать ход выполнения лабораторной работы с прокомментированными (назначение, описание параметров) листингами введенных команд IPTables, и команд, применяющихся для проверки конфигурации шлюза.

4.5 Контрольные вопросы

1. Опишите взаимосвязь основных компонентов netfilter. Возможно ли создание пользовательских таблиц.
2. Дайте определение терминального и нетерминального действия. Приведите пример отправки пакета в пользовательскую цепочку.
3. Какие цепочки содержатся в таблицах nat, mangle, filter, security, raw, rawpost. Опишите назначения цепочек INPUT, FORWARD, OUTPUT, PREROUTING, POSTROUTING.
4. Перечислите критерии состояния соединения. Опишите возможные состояния соединения.
5. Опишите действия ACCEPT, DROP, REJECT, RETURN, MARK, DELUDE. Все ли действия перечислены? Если нет, то опишите не перечисленные действия.
6. Назовите основные модули iptables. Как фильтровать пакеты по имени и UID программы отправившей пакет.
7. Опишите работу MASQUARADE при соединении хоста из локальной сети с хостом из внешней сети. Как «выглядит» соединение для внешнего хоста? Как маршрутизатор определяет адрес назначения пакета приходящего в ответ из внешней сети? Возможно ли установление соединения инициированого из внешней сети?

8. Опишите дополнительные критерии (multiport, iprange) и критерии маркировки (mark, connmark).
9. Опишите лимитирующие критерии (limit, hashlimit, connlimit, connbytes, quota, quota2, length, length2).
10. Опишите критерий recent. Приведите пример конфигурации для открытия порта «по стуку» при помощи критерия recent.

5 Самостоятельная работа №5. Настройка HTTP-прокси сервера Squid

5.1 Цель работы

Установка и конфигурирование HTTP proxy сервера Squid под операционной системой Linux. Конфигурирование клиентов виртуальной частной сети под операционными системами Linux и Windows.

5.2 Краткие теоретические сведения

Squid — программный пакет, реализующий функцию кэширующего прокси-сервера для протоколов HTTP, FTP, Gopher и (в случае соответствующих настроек) HTTPS. Разработан сообществом как программа с открытым исходным кодом (распространяется в соответствии с GNU GPL). Все запросы выполняет как один неблокируемый процесс ввода/вывода.

Используется в UNIX-like системах и в ОС семейства Windows NT. Имеет возможность взаимодействия с Active Directory Windows Server путём аутентификации через LDAP, что позволяет использовать разграничения доступа к интернет ресурсам пользователей, которые имеют учётные записи на Windows Server, также позволяет организовать «нарезку» интернет трафика для различных пользователей. Используется вместе с движками Mediawiki на wiki хостингах. Использование кэширующего прокси-сервера становится выгодно примерно с 2000 посетителей в сутки.

Сервер Squid развивается в течение уже многих лет. Обеспечивает совместимость с большинством важнейших протоколов Интернета, а также с операционными системами

Для контроля доступа к ресурсам и определения ряда действий используются списки контроля доступа. Каждый ACL может состоять из нескольких критериев (но только одного типа):

- адрес (сеть) источника запроса, цели запроса;
- имя (доменное имя) источника запроса, имя цели запроса;
- части URL запроса;
- протокол;
- порт (получателя, отправителя, самого squid’a);
- метод (POST или GET) при передаче данных по HTTP;
- браузер (User-agent);
- ident (запрос к рабочей станции);
- номер автономной системы отправителя/получателя (не для всех случаев);
- авторизация на прокси-сервере (см. ниже);
- номер соединения (чаще всего используется для ограничения количества соединений);
- SNMP;
- сертификаты пользователя;
- параметры запроса;

– внешние обработчики.

Squid поддерживает несколько видов идентификации пользователей:

– по IP-адресу (или доменному имени узла);
– по переданным реквизитам (логин/пароль);
– по идентификатору пользовательского агента (браузера).

Для идентификации по логину/паролю возможно использовать:

– обычные логин/пароль;
– NTLM-авторизацию;
– внешние программы авторизации (определяющие формат авторизации).

5.3 Ход работы

Вполне естественно, что адреса и некоторые другие аргументы команд в вашем случае будут другие. Не нужно перенабирать команды приведенные далее, необходимо разобраться для чего они предназначены и правильно использовать их для своих параметров сети.

Обратите внимание, что порядок разрешающих и запрещающих правил в squid.conf ИМЕЕТ ЗНАЧЕНИЕ.

Рекомендуется в качестве Host1 (рисунок 5.1) использовать хост на виртуальной машине к которому в настройках ВМ подключить 2 виртуальных сетевых адаптера в режиме моста.

Обратите внимание, что в VirtualBox может присутствовать ошибка, которая заключается в том, что выданный для одного из интерфейсов ВМ IP адрес обозначается роутером, как дубликат и пропадает подключение ко внешней сети. В случае возникновения данной проблемы необходимо вручную переназначить IP адрес проблемного интерфейса и исправить маршрутизацию.

5.3.1 Установка необходимых пакетов

В состав необходимых пакетов входят:

– squid (http://www.squid-cache.org/Versions/);
– freeradius, postgresql, установленные и настроенные в лабораторной работе №2.

Данные пакеты уже могут быть установлены в системе, в таком случае данный этап работы является не обязательным. В противном случае и в случае необходимости обновит уже установленные версии пакетов их необходимо загрузить из сети и установить. Существует несколько способов установки пакетов в систему: при помощи менеджера пакетов используемого дистрибутива Linux (apt-get, yum и т.д.) (загрузка и установка будут происходить автоматически); загрузка и установка уже собранного пакета для используемого дистрибутива Linux (deb, rpm и т.д.); загрузка исходного кода пакета с последующей его сборкой и установкой.

Рассмотрим наиболее универсальный вариант – установка пакетов из исходных кодов. Для этого необходимо загрузить исходные коды пакетов,

обычно помимо официального ресурса разработчика в сети существуют множество зеркал хранящих разные версии пакетов.

5.3.2 Загрузка пакетов

Пакеты необходимые для лабораторной работы можно загрузить с их официальных сайтов:

– squid (http://www.squid-cache.org/Versions/).

5.3.3 Распаковка пакетов

Разархивирование можно сделать командами:

```
$ tar xvf squid-<version>.tar.gz
```

В результате разархивирования должны быть созданы одноимённые с именами пакетов папки без префикса tar.gz.

5.3.4 Установка пакета squid

Войдите в корневую папку пакета squid-<version>. Для конфигурирования, компиляции и установки выполните следующие команды:

```
$ cd squid-<version>
$ ./configure --prefix=/ --exec-prefix=/usr --datarootdir=/usr/share --enable-basic-auth-helpers="squid_radius_auth" --with-logdir=/var/log/squid --with-default-user=squid --enable-delay-pools
$ make
$ sudo -s
# make install
# useradd -d /dev/null -M -s /bin/false squid
# mkdir -p /var/log/squid
# chown squid /var/log/squid
# squid -z
# squid
```

Проверьте работоспособность сервера:

```
# ps ax | grep squid
16666 ?        Ss     0:00 squid
16668 ?        S      0:00 (squid)
16689 pts/0    S+     0:00 grep squid
# netstat -nl | grep 3128
tcp6       0      0 :::3128                 :::*                    LISTEN
```

Более детальную инструкцию по установке squid из исходных кодов можно получить в файле squid-<version>/INSTALL.

5.3.5 Настройка проксирования HTTP трафика

Необходимо наличие сетевых двух интерфейсов с разными подсетями подключенными к ним. Пример схемы сети приведен на рисунке 5.1.

Проверьте доступность ресурсов внутренней сети политеха (cs.stu.cn.ua и stu.cn.ua) из виртуальной машины.

Для приведенной выше схемы сети, проводим первичную настройку сквида, убираем локальную сеть политеха из списка локальных сетей, и добавляем туда сеть 1.1.1.0/24 это необходимо для разрешения проксирования траффика из сети 1.1.1.0 в сеть 192.168.0.0, которую сквид будет считать внешней.

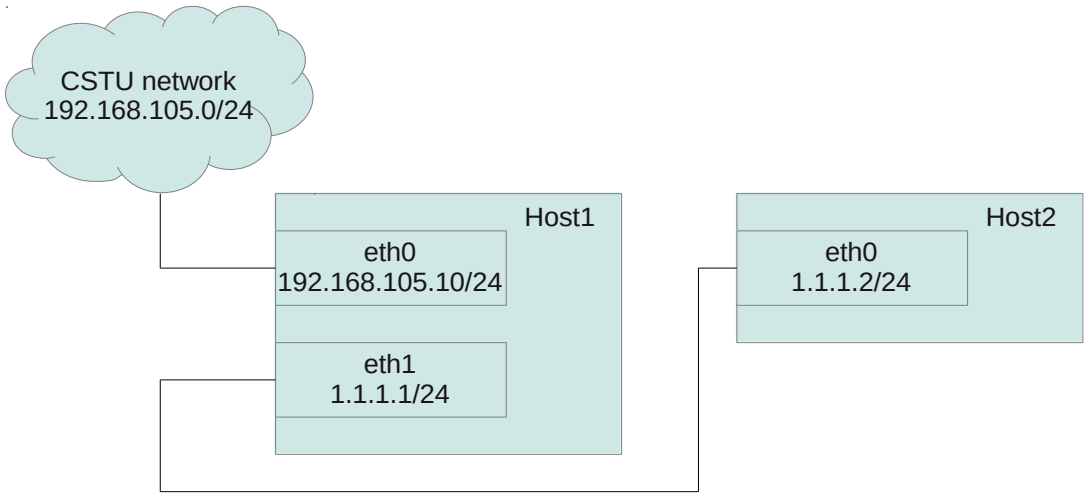

Рисунок 5.1 — Пример схемы сети

Пример конфигурационного файла /etc/squid.conf:

```
# Squid normally listens to port 3128
http_port 3128

#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

# Rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
acl localnet src 1.1.1.0/24     # RFC1918 possible internal network

#
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT

# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge

# Deny requests to unknown ports
http_access deny !Safe_ports

# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports

# Rule allowing access from your local networks.
```

```
http_access allow localnet

# And finally deny all other access to this proxy
http_access deny all

# Some security
hierarchy_stoplist cgi-bin ?

# Logging
access_log /var/log/squid/access.log squid
```

Перезагрузите конфигурацию сквид:

```
# squid -k reconfigure
```

Проверьте работы прокси-сервера из консоли:

```
$ export http_proxy="http://1.1.1.1:3128"
$ wget cs.stu.cn.ua
$ wget stu.cn.ua
```

После чего откройте загруженные файлы и убедитесь что там именно страница cs.stu.

5.3.6 Фильтрация по URL

Создайте файл со списком URL, которые будут блокироваться.
Пример файла /etc/pornlist:

```
stu.cn.ua
```

Если необходимо запретить все поддомены, то перед именем домена ставим точку.

Включите фильтр в squid (не забудьте перезагрузить конфигурацию).
Пример конфигурационного файла /etc/squid/squid.conf:

```
# Squid normally listens to port 3128
http_port 3128

#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

# Rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
acl localnet src 1.1.1.0/24     # RFC1918 possible internal network

#
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
```

```
acl CONNECT method CONNECT

# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge

# Deny requests to unknown ports
http_access deny !Safe_ports

# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports

# Deny acces to porn domains
acl PornSites dstdomain "/etc/pornlist"
http_access deny PornSites

# Rule allowing access from your local networks.
http_access allow localnet


# And finally deny all other access to this proxy
http_access deny all

# Some security
hierarchy_stoplist cgi-bin ?

# Logging
access_log /var/log/squid/access.log squid
```

Убедитесь, что URL не из файла /etc/pornlist доступны:

```
$ wget cs.stu.cn.ua
```

Проверьте что невозможно получить доступ к stu.cn.ua:

```
$ wget stu.cn.ua
```

5.3.7 Фильтрация по ключевым словам в URL

Пример конфигурационного файла /etc/squid/squid.conf

```
# Squid normally listens to port 3128
http_port 3128

#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

# Rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
acl localnet src 1.1.1.0/24     # RFC1918 possible internal network

#
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
```

```
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT

# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge

# Deny requests to unknown ports
http_access deny !Safe_ports

# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports

# Deny acces to porn domains
acl PornSites dstdomain "/usr/local/squid/etc/pornlist"
http_access deny PornSites

# Deny access to urls containing listed words
acl Porning url_regex pirate|post
http_access deny Porning

# Rule allowing access from your local networks.
http_access allow localnet


# And finally deny all other access to this proxy
http_access deny all

# Some security
hierarchy_stoplist cgi-bin ?

# Logging
access_log /var/log/squid/access.log squid
```

Убедитесь, что URL, не содержащие запрещенных слов доступны:

```
$ wget cs.stu.cn.ua
```

Проверьте что невозможно получить доступ, например, к посту на stu.cn.ua:

```
$ wget stu.cn.ua/post/744/
```

5.3.8 RADIUS авторизация

Сконфигурируйте подключение к радиус-серверу.

Пример конфигурационного файла /etc/squid_radius_config:

```
server 127.0.0.1
secret testing123
```

Запустите серверы postgres и freeradius как указано в лабораторной работе №2. Проверьте настройки подключения и авторизации при помощи запуска модуля авторизации, после чего введите имя пользователя и пароль через пробел и нажимаем Enter:

```
$ /usr/libexec/squid_radius_auth -f /etc/squid_radius_config
linux test123
OK
```

Если появилось сообщение OK значит все работает нормально завершите работу приложения нажатием Ctrl+C.

Настройте аутентификацию в squid.
Пример конфигурационного файла /etc/squid/squid.conf:

```
# Squid normally listens to port 3128
http_port 3128

#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

# Rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
acl localnet src 1.1.1.0/24     # RFC1918 possible internal network

#
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT

# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge

# Deny requests to unknown ports
http_access deny !Safe_ports

# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports


# Deny acces to porn domains
acl PornSites dstdomain "/usr/local/squid/etc/pornlist"
http_access deny PornSites

# Deny access to urls containing listed words
acl Porning url_regex pirate|post
http_access deny Porning

# Rule allowing access only from your local networks.
http_access deny !localnet

# Radius auth
auth_param basic program /usr/local/squid/libexec/squid_radius_auth -f
/etc/squid/radius_config
auth_param basic children 5
auth_param basic realm Web-Proxy
auth_param basic credentialsttl 5 minute
auth_param basic casesensitive off

acl radius-auth proxy_auth REQUIRED
http_access allow radius-auth

# Rule allowing access from your local networks.
```

```
#http_access allow localnet

# And finally deny all other access to this proxy
http_access deny all

# Some security
hierarchy_stoplist cgi-bin ?

# Logging
access_log /var/log/squid/access.log squid
```

Проверьте работы прокси-сервера из консоли:

```
$ export http_proxy="http://linux:test123@1.1.1.1:3128"
$ wget cs.stu.cn.ua
```

После чего откройте загруженный файл и убедитесь что там именно страница cs.stu.

Проверьте также вариант, когда имя пользователя или пароль заданы неверно.

5.3.9 Ограничение пропускной способности

Задайте ограничение скорости для локальной подсети localnet.

Пример конфигурационного файла /etc/squid/squid.conf:

```
# Squid normally listens to port 3128
http_port 3128

#Recommended minimum configuration:
acl all src all
acl manager proto cache_object
acl localhost src 127.0.0.1/32
acl to_localhost dst 127.0.0.0/8 0.0.0.0/32

# Rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
acl localnet src 10.0.0.0/8     # RFC1918 possible internal network
acl localnet src 172.16.0.0/12  # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
acl localnet src 1.1.1.0/24     # RFC1918 possible internal network

#
acl SSL_ports port 443          # https
acl SSL_ports port 563          # snews
acl SSL_ports port 873          # rsync
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443         # https
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 631         # cups
acl Safe_ports port 873         # rsync
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT

# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager

# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge

# Deny requests to unknown ports
http_access deny !Safe_ports
```

```
        # Deny CONNECT to other than SSL ports
        http_access deny CONNECT !SSL_ports


        # Deny acces to porn domains
        acl PornSites dstdomain "/usr/local/squid/etc/pornlist"
        http_access deny PornSites

        # Deny access to urls containing listed words
        acl Porning url_regex pirate|post
        http_access deny Porning

        # Rule allowing access only from your local networks.
        http_access deny !localnet

        # Radius auth
        auth_param basic program /usr/local/squid/libexec/squid_radius_auth -f
/etc/squid/radius_config
        auth_param basic children 5
        auth_param basic realm Web-Proxy
        auth_param basic credentialsttl 5 minute
        auth_param basic casesensitive off

        acl radius-auth proxy_auth REQUIRED
        http_access allow radius-auth

        # And finally deny all other access to this proxy
        http_access deny all

        # Some security
        hierarchy_stoplist cgi-bin ?

        # Logging
        access_log /var/log/squid/access.log squid

        # DELAY POOL PARAMETERS (all require DELAY_POOLS compilation option)
        delay_pools 3
        delay_class 1 2
        delay_class 2 2
        delay_class 3 2

        delay_access 3 deny all
        delay_access 2 allow localnet
        delay_access 2 deny all
        delay_access 1 deny all

        delay_parameters 1 3000/3000 1500/1500
        delay_parameters 2 -1/-1 6000/6000
        delay_parameters 3 -1/-1 -1/-1
```

Еще раз загрузите страницу и сравните скорость загрузки и время потраченное на загрузку страницы в сравнении с предыдущим разом:

```
$ wget cs.stu.cn.ua
```

Самостоятельно проверьте скорости для двух других пулов заданных в примере.

5.4 Содержание отчета

Отчёт должен содержать ход выполнения лабораторной работы с прокомментированными изменениями конфигурационных файлов Squid, а также листинг команд, применяющихся для проверки конфигурации сервера.

5.5 Контрольные вопросы

1. Что такое списки контроля доступа в Squid? Перечислите основные критерии используемые при формировании списков контроля доступа в Squid. Приведите пример списка контроля доступа для запрета доступа к серверам из определенной подсети?

2. Перечислите виды идентификации в Squid. Назовите методы идентификации по логину/паролю поддерживаемые Squid?
3. Поддерживает ли Squid SSL/HTTPS/TLS? Как браузер туннелирует запросы через Squid?
4. Приведите пример конфигурации Squid обеспечивающий прямой доступ к некоторым сайтам.
5. Приведите пример конфигурации Squid запрещающий закачивание больших файлов.
6. Назовите и объясните назначение каждого из приведенных в лабораторной работе параметров конфигурации Squid для ограничения скорости.
7. Что такое «HTTP_X_FORWARDED_FOR»? Почему Squid снабжает им WWW-сервера? Приведите пример конфигурации запрещающий добавление «HTTP_X_FORWARDED_FOR».
8. Опишите основные принципы анонимизации HTTP-запросов? Как это связано с «HTTP_X_FORWARDED_FOR»? Приведите пример конфигурации Squid запрещающий определенные заголовки.
9. Опишите базовую конфигурацию Squid для работы в режиме прозрачного прокси.
10. Опишите конфигурацию Squid при которой все запросы перенаправляются другому прокси-серверу если он доступен иначе запросы отправляются напрямую.

Рекомендованная литература

1. Carla Schroder. Linux Networking Cookbook. M.: O'Reilly Media, 2007.
2. Tony Bautts, Terry Dawson, Gregor N. Purdy. Linux Network Administrator's Guide. M.: O'Reilly Media, 2005.
3. Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley. UNIX and Linux System Administration Handbook (4th Edition). M.: Prentice Hall, 2010.
4. James Turnbull. Hardening Linux. M.: Apress, 2005.
5. Ed Sawicki. Advanced Guide to Linux Networking and Security. M.: Course Technology, 2005.
6. Mike Erwin, Charlie Scott, Paul Wolfe. Virtual Private Networks, 2nd Edition. M.: O'Reilly Media, 1998.
7. Oleg Kolesnikov, Brian Hatch. Building Linux Virtual Private Networks (VPNs). M.: Sams, 2002.
8. Jonathan Hassell. RADIUS. Securing Public Access to Private Resources. M.: O'Reilly Media, 2002.
9. Michael Rash. Linux Firewalls: Attack Detection and Response with iptables, psad, and fwsnort. M.: No Starch Press, 2007.
10. Steve Suehring, Robert Ziegler. Linux Firewalls (3rd Edition). M.: Novell Press, 2005.
11. Kyle Dent D. Postfix: The Definitive Guide. M.: O'Reilly Media, 2003.
12. Don R Crawley. The Accidental Administrator: Linux Server Step-by-Step Configuration Guide. M.: CreateSpace, 2010.
13. Alistair McDonald. SpamAssassin: A Practical Guide to Integration and Configuration. M.: Packt Publishing, 2004.
14. Alan Schwartz PH.D. SpamAssassin. M.: O'Reilly Media, 2004.
15. Duane Wessels. Web Caching. M.: O'Reilly Media, 2001.
16. Ari Luotonen. Web Proxy Servers. M.: Prentice Hall PTR, 1997.
17. Duane Wessels. Squid: The Definitive Guide. M.: O'Reilly Media, 2004.