1. ОРГАНИЗАЦИОННО-МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ

1.1. Цель и задачи выполнения практических занятий

В соответствии с учебным планом студентов специальности 230101 и рабочей программой по дисциплине «Сети ЭВМ и телекоммуникации» и изложенными в них требованиями к уровню подготовки инженеров для работы в организациях ГА студенты должны обладать практическими навыками в решении задач, связанных с использованием различных технологий и протоколов межсетевого взаимодействия.

Целью данного пособия является закрепление студентами теоретического курса дисциплины и приобретение навыков настройки и использования утилит TCP/IP службы DHCP, Proxy, протоколов PPTP.

1.2. Основные вопросы, подлежащие изучению:

1) логика работы протоколов стека TCP/IP;
2) освоение приемов и методов работы со службой динамического выделения адресов DHCP;
3) изучение технологии работы Proxy-сервера;
4) освоение приемов и методов работы с протоколами создания виртуальных частных сетей.

2. ПЕРЕЧЕНЬ ТЕМ ПРАКТИЧЕСКИХ ЗАНЯТИЙ

Учебным планом предусмотрены практические занятия в объеме 8 часов:

1. Конфигурирование службы DHCP в корпоративной сети (2 часа).
2. Утилиты TCP/IP в среде Windows (2 часа).
3. Установка и администрирование Proxy-сервера (2 часа).
4. Создание виртуальной частной сети на базе PPTP сервера POPTOP (2часа).

3. СОДЕРЖАНИЕ ЗАНЯТИЙ

3.1. КОНФИГУРИРОВАНИЕ СЛУЖБЫ DHCP В КОРПОРАТИВНОЙ СЕТИ

3.1.1. Цель занятия:

- ознакомление с сервисом DHCP;
- приобретение навыков в настройке сервиса DHCP.

3.1.2. Методические указания по теме

Общие сведения

Dynamic Host Configuration Protocol (DHCP) – протокол динамической конфигурации хостов предназначен для автоматической настройки параметров стека TCP/IP рабочей станции в момент ее загрузки. Он используется для настройки изменяемых сетевых параметров хостов (клиентов) с помощью сервера. Настраиваются следующие основные параметры: IP-адрес и маска сетевого интерфейса, маршрут по умолчанию, адреса серверов DNS и WINS в сети.

Станция во время загрузки, или точнее во время активации сетевого интерфейса, выдаёт широковещательный запрос параметров своей конфигурации. Сервер DHCP откликнется на этот запрос по адресу запросившей станции и предоставит ей конфигурационные данные.

Процесс взаимодействия сервера и клиента происходит в следующем порядке. Сервер получает запрос и откликается с предложением об аренде (lease), содержащим конфигурационные данные для хоста. Ресурс, содержащийся в предложении сервера, временно блокируется для предложения другим хостам до получения ответа от хоста или истечения тайм-аута. Хост может получить предложения от нескольких DHCP-серверов, работающих в данном широковещательном сегменте сети. Хост, на основании настроек своего DHCP-клиента, решает принять предложение определенного сервера (или принять первое поступившее предложение, если никаких настроек нет). Хост отвечает выбранному серверу сообщением "выбор". Сервер подтверждает выдачу аренды; после получения подтверждения хост конфигурирует себя в соответствии с полученными данными.

Один DHCP-сервер может работать в нескольких сетях. Для этого в каждой сети должен быть доступен сконфигурированный DHCP-relay – специальный посредник, который будет ретранслировать сообщения между сервером и хостом, запросившим конфигурацию. Без посредника DHCP-сервер не услышит запросов, так как широковещательные IP-дейтаграммы не выходят за пределы сегмента сети.

IP-адрес, присваиваемый рабочей станции, может выдаваться сервером из пространства специально для этого выделенных адресов (берется первый свободный адрес). В этом случае у рабочей станции нет постоянного IP-адреса. Этот вариант приемлем для мобильных клиентов в местах общего доступа к сети.

IP-адрес, присваиваемый конкретной рабочей станции, может быть и фиксированным, для этого надо указать MAC-адрес (Ethernet-адрес) рабочей станции и IP-адрес в настройках сервера. Последний вариант является более предпочтительным в корпоративных сетях из соображений безопасности сети, поскольку всегда можно однозначно идентифицировать, с какого хоста производятся те или иные действия и, с другой стороны, выдавать параметры конфигурации только хостам с известными MAC-адресами.

В любом случае использование DHCP позволяет избежать конфигурирования стека TCP/IP на каждом хосте сети отдельно и проводить гибкую, централизованную административную политику.

Настройка DHCP-сервера Windows

Windows XP имеет поставляемый с системой сервер DHCP. Для работы этого сервера необходимо:

- в настройках сети (Настройки – Панель управления – Сеть), в разделе Services добавить Microsoft DHCP Server;
- запустить сервер через Control Panel – Services – DHCP Server кнопкой Start;
- сервер настраивается с помощью программы DHCP Manager, запускаемой из раздела Administrative Tools.

Для каждого из серверов (программа позволяет управлять несколькими серверами) существует один или несколько контекстов (scope), описывающих конфигурацию и настройки сервера для той или иной сферы действия. В простейшем случае имеется один сервер с одним контекстом. Серверы и их контексты показываются в левой части окна программы.

Если контекста нет, его следует создать через меню Scope-Create. Существующий контекст можно редактировать через меню Scope-Properties. В конфигурации контекста указывается диапазон IP-адресов, выделенный для динамического распределения адресов для клиентов, а также поддиапазоны, которые следует исключить (exclude) из этого диапазона. Параметр Lease Duration указывает максимальную продолжительность использования IP-адреса клиентом; значение Unlimited определяет неограниченное время использования.

Меню Scope-Reservations позволяет зафиксировать IP-адреса за определенными хостами (точнее, за определенными Ethernet-адресами). Ethernet-адрес указывается в поле Unique Identifier.

Передача клиентам дополнительной информации (адрес шлюза, адрес DNS-сервера и доменное имя и т.п.) конфигурируется через меню DHCP Options (Global – для всех контекстов, Scope – для данного контекста). Выберите нужные опции, активизируйте их с помощью кнопки Add и укажите значения требуемых параметров для каждой опции.

Опции для клиентов с фиксированными адресами устанавливаются через меню Scope-Active Leases, далее двойным щелчком вызвать свойства нужного клиента.

Для ввода контекста в действие используйте меню Scope-Activate (Deactivate – для отключения контекста).

Настройка DHCP-клиента Windows

DHCP-клиент под Windows активизируется через *Настройки – Панель управления – Сеть – TCP/IP – Свойства – IP-адрес – Получить IP-адрес автоматически.* Если на хосте не сконфигурированы параметры DNS и адрес

шлюза, они будут получены от DHCP-сервера, иначе будут использоваться уже имеющиеся настройки.

В случае отсутствия DHCP-сервера в сети при включенном автоматическом получении IP-адреса хост присвоит себе адрес самостоятельно. В этом случае возможно отсутствие коннективности из-за некорректного адреса.

Настройка DHCP-сервера Linux

В лабораторных работах используется DHCP-сервер, разработанный Internet Software Consortium (http://www.isc.org).

Функции DHCP сервера выполняет демон dhcpd, конфигурация которого описывается в файле /etc/dhcpd.conf. В файл /var/db/dhcpd/dhcpd.leases сервер заносит информацию о выделенных адресах. Для работы с сервером необходимо создать конфигурационный файл, после чего запустить программу-демон.

В конфигурационном файле определяются пространства IP-адресов, назначаемых клиентам, дополнительная информация по конфигурации стека TCP/IP, передаваемая клиентам, а также описываются хосты, которым назначаются фиксированные IP-адреса (по MAC-адресу хоста). В начале файла можно указать глобальные опции, передаваемые всем клиентам.

Далее для каждой обслуживаемой сервером IP-сети создается отдельный раздел, где указываются:

- маска сети (netmask);
- диапазон(ы) выдаваемых IP-адресов (range);
- время по умолчанию, на которое выдается адрес, в секундах (default-lease-time);
- максимальное время, на которое может быть выдан адрес, если хост запрашивает конкретное время, в секундах (max-lease-time);
- дополнительные опции (option), передаваемые клиентам, например:
- маска сети, передаваемая клиенту (subnet-mask);
- широковещательный адрес (broadcast-address);
- адреса шлюзов (для маршрута по умолчанию) (routers);
- имя домена (domain-name);
- адрес сервера WINS;
- адреса DNS-серверов (domain-name-servers).

Если какая-либо из опций уже определена глобально, то локальная опция заменяет значение глобальной опции для данной сети.

Пример конфигурации для обслуживаемой сети:

```
default-lease-time 86400;
max-lease-time 604800;
get-lease-hostnames true;
option subnet-mask 255.255.255.192;          ;маска подсети
option domain-name "stu";                    ;имя домена
option domain-name-servers 192.168.0.10      ;IP-адрес сервера доменов
```

```
option interface-mtu 1500;
ddns-update-style none;                    стиль динамического обновления DNS
server-name dhcp-server-73-1;

subnet 192.168.7.0 netmask 255.255.255.192 {
option routers 192.168.7.1
option broadcast-address 192.168.7.63;
; диапазон выдаваемых адресов
  range 192.168.7.30 192.168.7.50
}
```

Для каждого из хостов, которым выдается фиксированный адрес, создается отдельный раздел с заголовком "host hostname", где hostname – имя хоста. Внутри раздела указываются MAC-адрес хоста (в случае Ethernet: hardware ethernet address) и IP-адрес, выдаваемый хосту (fixed-address IP-address). Также могут указываться опции такие же, как и для сети. Если опции не указаны, хосту будут переданы опции, определенные в разделе конфигурации сети, в которой находится хост, или глобальные опции, в порядке приоритета.

Пример раздела конфигурации хоста:

```
host ics-73-5 {
hardware ethernet 00:50:BA:57:79:4E;
fixed-address 192.168.7.19;
}
```

Хосты можно объединять в группы, с указанием опций, общих для всех хостов данной группы, перед разделами с описанием хостов:

```
group {
option domain-name-servers stalker.stu;
host ics-73-5  {
    ...
  }
  host ics-73-6 {
    ...
  }
}
```

Запуск программы dhcpd может осуществляться в файле начальной загрузки типа /etc/rc/* (детали зависят от вида операционной системы). Некоторые параметры командной строки:

dhcpd [-p port] [-cf configfile] [if0 [...ifN]] ,

где port – номер UDP порта, если он отличается от стандартного (67); configfile – имя конфигурационного файла, если это не ./dhcpd.conf; if0 ... ifN – сетевые интерфейсы, обслуживаемые демоном (если у хоста несколько интерфейсов).

Настройка DHCP-клиента Linux

DHCP-клиент ОС под Unix из пакета Internet Software Consortium DHCP состоит из программы dhclient, конфигурационного файла /etc/dhclient.conf и файла dhclient.leases в который клиент заносит информацию о выданных ему адресах и настройках. Для запуска клиента во время загрузки системы используется специальный скрипт (сценарий оболочки), обычно встроенный в скрипт активации сетевого интерфейса.

Конфигурационный файл в большинстве случаев очень прост и часто он даже может быть пуст. Ниже приведен ряд полезных директив конфигурационного файла dhclient.conf.

timeout time:

если через time секунд ответ от сервера не получен, хост пытается конфигурироваться самостоятельно, используя информацию о предыдущих конфигурациях из файла dhclient.leases (если их срок годности не истек) или используя статически установленные конфигурации; каждая такая конфигурация-кандидат проверяется на работоспособность. Формат записи конфигураций – см. man dhclient.conf. В случае неудачи попытка соединения с сервером повторяется в соответствии с параметром retry; значение timeout по умолчанию – 60 с;

retry time:

период повторных попыток соединения с сервером в случае неудачи; измеряется в секундах, по умолчанию – 300 с;

request option:

запросить у сервера передачу опции option;

require option:

в случае если сервер не передал опцию option, отвергнуть конфигурацию, предложенную сервером;

send option declaration:

передать серверу значение declaration опции option, например:

send requested-lease-time 7200:

запросить выделение IP-адреса на 7200 с;

default option declaration:

установить значение declaration для опции option, если сервер не передал эту опцию;

supersede option declaration:

установить значение declaration для опции option, независимо от того, что передал сервер;

prepend option declaration:

добавить значение для опции к значению, переданному сервером, поставив свое значение первым;

append option declaration:

добавить значение для опции к значению, переданному сервером, поставив свое значение последним.

Директивы prepend и append должны использоваться только для опций, допускающих множественные значения, иначе результат получится непредсказуемым.

reject ip_address:

не принимать предложения от DHCP-сервера, который идентифицирует себя адресом ip_address;

interface "if_name" { директивы }:

если у компьютера несколько интерфейсов, директивы в разделе interface будут относиться к конфигурации интерфейса if_name. Интерфейсы, не имеющие соответствующих разделов в конфигурационном файле, будут конфигурироваться с учетом глобальных директив или по умолчанию.

3.1.3. Задания на выполнение практической работы

1. Настроить DHCP-сервер в среде Windows и проверить его работу на клиентских машинах, используя DHCP-клиент.
2. Настроить DHCP-сервер в среде Linux и проверить его работу на клиентских машинах, используя DHCP-клиент.

3.1.4. Контрольные вопросы

1. Что такое DHCP? Необходим ли подобный сервис в локальных сетях?
2. Зачем нужен dhclient?
3. Как сконфигурировать DHCP-сервер под Unix?
4. Как сконфигурировать DHCP-клиент под Unix?
5. Что такое диапазон адресов (range)?
6. Опишите механизм выделения IP-адресов с помощью сетевого сервиса DHCP.
7. В каком случае рекомендуется выделять фиксированные адреса хостов?
8. Какие параметры получает рабочая станция от сервера DHCP?

3.2. УТИЛИТЫ TCP/IP В СРЕДЕ WINDOWS

3.2.1. Цель занятия

Практически освоить работу с утилитами TCP/IP.

3.2.2. Методические указания по теме

Диагностические утилиты TCP/IP

В состав TCP/IP входят диагностические утилиты, предназначенные для проверки конфигурации стека и тестирования сетевого соединения:

arp

выводит для просмотра и изменения таблицу трансляции адресов, используемую протоколом разрешения адресов ARP (Address Resolution Protocol – определяет локальный адрес по IP-адресу);

hostname

выводит имя локального хоста. Используется без параметров;

ipconfig

выводит значения для текущей конфигурации стекаTCP/IP: IP-адрес, маску подсети, адрес шлюза по умолчанию, адреса WINS (Windows Internet Naming Service) и DNS (Domain Name System);

nbtstat

выводит статистику и текущую информацию по NetBIOS, установленному поверх TCP/IP. Используется для проверки состояния текущих соединений NetBIOS;

netstat

выводит статистику и текущую информацию по соединению TCP/IP;

nslookup

осуществляет проверку записей и доменных псевдонимов хостов, доменных сервисов хостов, а также информации операционной системы, путем запросов к серверам DNS;

ping

осуществляет проверку правильности конфигурирования TCP/IP и проверку связи с удаленным хостом;

route

модифицирует таблицы маршрутизации IP. Отображает содержимое таблицы, добавляет и удаляет маршруты IP;

tracert

осуществляет проверку маршрута к удаленному компьютеру путем отправки эхо-пакетов протокола ICMP (Internet Control Message Protocol). Выводит маршрут прохождения пакетов на удаленный компьютер.

Проверка правильности конфигурации TCP/IP

При устранении неисправностей и проблем в сети TCP/IP следует сначала проверить правильность конфигурации TCP/IP. Для этого используется утилита ipconfig.

Эта команда полезна на компьютерах, работающих с DHCP (Dynamic Host Configuration Protocol), так как дает пользователям возможность определить, какая конфигурация сети TCP/IP и какие величины были установлены с помощью DHCP.

Синтаксис:

ipconfig [/all | /renew[adapter] | /release]

Параметры:

all

выдает весь список параметров. Без этого ключа отображается только IP-адрес, маска и шлюз по умолчанию;

renew[adapter]

обновляет параметры конфигурации DHCP для указанного сетевого адаптера;

release[adapter]

освобождает выделенный DHCP IP-адрес; *adapter* – имя сетевого адаптера;

displaydns

выводит информацию о содержимом локального кэша клиента DNS, используемого для разрешения доменных имен.

Таким образом, утилита ipconfig позволяет выяснить, инициализирована ли конфигурация и не дублируются ли IP-адреса:

- если конфигурация инициализирована, то появляется IP-адрес, маска, шлюз;
- если IP-адреса дублируются, то маска сети будет 0.0.0.0;
- если при использовании DHCP компьютер не смог получить IP-адрес, то он будет равен 0.0.0.0.

Тестирование связи с использованием утилиты ping

Утилита ping (Packet Internet Grouper) используется для проверки конфигурирования TCP/IP и диагностики ошибок соединения. Она определяет доступность и функционирование конкретного хоста. Использование ping лучший способ проверки того, что между локальным компьютером и сетевым хостом существует маршрут. Хостом называется любое сетевое устройство (компьютер, маршрутизатор), обменивающееся информацией с другими сетевыми устройствами по TCP/IP.

Команда ping проверяет соединение с удаленным хостом путем посылки к этому хосту эхо-пакетов ICMP и прослушивания эхо-ответов. Ping ожидает каждый посланный пакет и печатает количество переданных и принятых пакетов. Каждый принятый пакет проверяется в соответствии с переданным сообщением. Если связь между хостами плохая, из сообщений ping станет ясно, сколько пакетов потеряно.

По умолчанию передается 4 эхо-пакета длиной 32 байта (периодическая последовательность символов алфавита в верхнем регистре). Ping позволяет изменить размер и количество пакетов, указать, следует ли записывать маршрут, который она использует, какую величину времени жизни (ttl) устанавливать, можно ли фрагментировать пакет и т.д. При получении ответа в поле time указывается, за какое время (в миллисекундах) посланный пакет доходит до удаленного хоста и возвращается назад. Так как значение по умолчанию для ожидания отклика равно 1 с, то все значения данного поля будут меньше 1000 мс. Если вы получаете сообщение «Request time out» (Превышен интервал ожидания), то, возможно, если увеличить время ожидания отклика, пакет дойдет до удаленного хоста. Это можно сделать с помощью ключа –w.

Ping можно использовать для тестирования как имени хоста (DNS или NetBIOS), так и его IP-адреса. Если команда ping с IP-адресом выполнилась успешно, а с именем – неудачно, это значит, что проблема заключается в распознавании соответствия адреса и имени, а не в сетевом соединении.

Утилита ping используется следующими способами:

1. Для проверки того, что TCP/IP установлен и правильно сконфигурирован на локальном компьютере, в команде ping задается адрес петли обратной связи (loopback address):

ping 127.0.0.1

Если тест успешно пройден, то вы получите следующий ответ:

Reply from 127.0.0.1
Reply from 127.0.0.1
Reply from 127.0.0.1
Reply from 127.0.0.1

2. Чтобы убедиться в том, что компьютер правильно добавлен в сеть и IP-адрес не дублируется, используется IP-адрес локального компьютера:

ping IP-адрес_локального_хоста

3. Чтобы проверить, что шлюз по умолчанию функционирует и что можно установить соединение с любым локальным хостом в локальной сети, задается IP-адрес шлюза по умолчанию:

ping IP-адрес_шлюза

4. Для проверки возможности установления соединения через маршрутизатор в команде ping задается IP-адрес удаленного хоста:

ping IP-адрес_удаленного хоста

Синтаксис утилиты ping:

```
ping  [-t] [-a] [-n count] [-l length] [-f] [-i ttl]
[-v tos] [-r count] [-s count] [ [-j host-list] |
[-k host-list] ] [-w timeout]   destination-list
```

Параметры:

-t

выполняет команду ping до прерывания. Control-Break – посмотреть статистику и продолжить. Control-C – прервать выполнение команды;

-a

позволяет определить доменное имя удаленного компьютера по его IP-адресу;

-n count

посылает количество пакетов ECHO, указанное параметром count;

-l length

посылает пакеты длиной length байт (максимальная длина 8192 байта);

-f

посылает пакет с установленным флагом «не фрагментировать». Этот пакет не будет фрагментироваться на маршрутизаторах по пути своего следования;

-i ttl

устанавливает время жизни пакета в величину ttl (каждый маршрутизатор уменьшает ttl на единицу);

-v tos

устанавливает тип поля «сервис» в величину tos;

-r count

записывает путь выходящего пакета и возвращающегося пакета в поле записи пути. Count – от 1 до 9 хостов;

-s count

позволяет ограничить количество переходов из одной подсети в другую (хопов). Count задает максимально возможное количество хопов;

-j host-list

направляет пакеты с помощью списка хостов, определенного параметром host-list. Последовательные хосты могут быть отделены промежуточными маршрутизаторами (гибкая статическая маршрутизация). Максимальное количество хостов в списке, позволенное IP, равно 9;

-k host-list

направляет пакеты через список хостов, определенный в host-list. Последовательные хосты не могут быть разделены промежуточными маршрутизаторами (жесткая статическая маршрутизация). Максимальное количество хостов – 9;

-w timeout

указывает время ожидания (timeout) ответа от удаленного хоста в миллисекундах (по умолчанию – 1 с);

destination-list

указывает удаленный хост, к которому надо направить пакеты ping.

Пример использования утилиты ping

```
C:\WINDOWS>ping –n 10 www.netscape.com
Обмен пакетами с www.netscape.com [205.188.247.65] по 32 байт:
Ответ от 205.188.247.65: число байт=32 время=194мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=240мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=173мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=250мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=187мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=239мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=263мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=230мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=185мс TTL=48
Ответ от 205.188.247.65: число байт=32 время=406мс TTL=48
```

Статистика Ping для 205.188.247.65
Пакетов: послано = 10, получено = 10, потеряно = 0 (0% потерь)
Приблизительное время передачи и приема
Наименьшее = 173мс, наибольшее = 406мс, среднее =236мс

Изучение маршрута между сетевыми соединениями с помощью утилиты tracert

Tracert – это утилита трассировки маршрута. Она использует поле TTL (time-to-live, время жизни) пакета IP и сообщения об ошибках ICMP для определения маршрута от одного хоста до другого.

Утилита tracert может быть более содержательной и удобной, чем ping, особенно в тех случаях, когда удаленный хост недостижим. С помощью нее можно определить район проблем со связью (у Internet-провайдера, в опорной сети, в сети удаленного хоста) по тому, насколько далеко будет отслежен маршрут. Если возникли проблемы, то утилита выводит на экран звездочки (*), либо сообщения типа «Destination net unreachable», «Destination host unreachable», «Request time out», «Time Exeeded».

Утилита tracert работает следующим образом: посылается по три пробных эхо-пакета на каждый хост, через который проходит маршрут до удаленного хоста. На экран при этом выводится время ожидания ответа на каждый пакет (его можно изменить с помощью параметра -w). Пакеты посылаются с различными величинами времени жизни. Каждый маршрутизатор, встречающийся по пути, перед перенаправлением пакета уменьшает величину TTL на единицу. Таким образом, время жизни является счетчиком точек промежуточной доставки (хопов). Когда время жизни пакета достигнет нуля, предполагается, что маршрутизатор пошлет в компьютер-источник сообщение ICMP “Time Exeeded” (“время истекло”). Маршрут определяется путем посылки первого эхо-пакета с TTL=1. Затем TTL увеличивается на 1 в каждом последующем пакете до тех пор, пока пакет не достигнет удаленного хоста, либо будет достигнута максимально возможная величина TTL (по умолчанию 30, задается с помощью параметра -h).

Маршрут определяется путем изучения сообщений ICMP, которые присылаются обратно промежуточными маршрутизаторами.

Примечание: некоторые маршрутизаторы просто молча уничтожают пакеты с истекшим TTL и не будут видны утилите tracert.

Синтаксис:

tracert [-d] [-hmaximum_hops] [-jhost-list] [-wtimeout] имя_целевого_хоста

Параметры:

-d

указывает, что не нужно распознавать адреса для имен хостов;

-h maximum_hops

указывает максимальное число хопов для того, чтобы искать цель;

-j host-list

указывает нежесткую статическую маршрутизацию в соответствии с host-list;

-w timeout

указывает, что нужно ожидать ответ на каждый эхо-пакет заданное число мс.

Утилита ARP

Основная задача протокола ARP – трансляция IP-адресов в соответствующие локальные адреса. Для этого ARP-протокол использует информацию из ARP-таблицы (ARP-кэша). Если необходимая запись в таблице не найдена, то протокол ARP отправляет широковещательный запрос ко всем компьютерам локальной подсети, пытаясь найти владельца данного IP-адреса. В кэше могут содержаться два типа записей: статические и динамические. Статические записи вводятся вручную и хранятся в кэше постоянно. Динамические записи помещаются в кэш в результате выполнения широковещательных запросов. Для них существует понятие времени жизни. Если в течение определенного времени (по умолчанию 2 мин.) запись не была востребована, то она удаляется из кэша.

Синтаксис:

arp [-s inet_addr eth_addr] | [-d inet_addr] | [-a]

Параметры:

-s

занесение в кэш статических записей;

-d

удаление из кэша записи для определенного IP-адреса;

-a

просмотр содержимого кэша для всех сетевых адаптеров локального компьютера;

inet_addr – IP-адрес;

eth_addr – MAC-адрес.

Утилита netstat

Утилита netstat позволяет получить статическую информацию по некоторым из протоколов стека (TCP, UDP, IP, ICMP), а также выводит сведения о текущих сетевых соединениях. Особенно она полезна на брандмауэрах, с ее помощью можно обнаружить нарушения безопасности периметра сети.

Синтаксис:

netstat [-a] [-e] [-n] [-s] [-p protocol] [-r]

Параметры:

-a

выводит перечень всех сетевых соединений и прослушивающихся портов локального компьютера;

-e

выводит статистику для Ethernet-интерфейсов (например, количество полученных и отправленных байт);

-n

выводит информацию по всем текущим соединениям (например, TCP) для всех сетевых интерфейсов локального компьютера. Для каждого соединения выводится информация об IP-адресах локального и удаленного интерфейсов вместе с номерами используемых портов;

-s

выводит статистическую информацию для протоколов UDP, TCP, ICMP, IP. Ключ «/more» позволяет просмотреть информацию постранично;

-r

выводит содержимое таблицы маршрутизации.

3.2.3. Задания на выполнение практической работы

1. Получить справочную информацию по командам ipconfig, ping, tracert, hostname.
2. Получить имя хоста.
3. Изучить утилиты ipconfig.
4. Протестировать связь с помощью утилиты ping.
5. Определить путь IP-пакета.
6. Просмотреть ARP-кэш.
7. Получить информацию о текущих сетевых соединениях и протоколах стека TCP/IP.

3.2.4. Контрольные вопросы

1. Какие утилиты можно использовать для проверки правильности конфигурирования TCP/IP?
2. Каким образом команда ping проверяет соединение с удаленным хостом?
3. Что такое хост?
4. Что такое петля обратной связи?
5. Сколько промежуточных маршрутизаторов сможет пройти IP-пакет, если его время жизни равно 30?
6. Как работает утилита tracert?
7. Каково назначение протокола ARP?

3.3. PROXY-СЕРВЕРЫ

3.3.1. Цель занятия

Настройка и администрирование Proxy-сервера **squid** под операционной системой Linux.

3.3.2. Методические указания по теме

Proxy-сервер, осуществляющий доступ в Internet, предоставляет следующие возможности:

- централизованный выход в Internet через один сервер в сети;
- локальное хранение часто просматриваемых документов для увеличения скорости загрузки страниц (один пользователь загрузил документ с удаленного сервера в Internet, а все остальные после этого берут этот документ с Proxy-сервера);
- регулирование пропускной способности канала в зависимости от его нагрузки;
- авторизованный доступ в Internet (пользователь может загружать документы из Internet только при наличии логина и пароля).

Установка Proxy-сервера

Прежде чем устанавливать Proxy-сервер s*quid*, надо убедиться в том, что:

- машина, на которой будет работать *Proxy*, может соединиться (по WWW, FTP, Telnet – неважно) с другими машинами в сети;
- машины из внутренней сети могут соединиться с машиной, на которую устанавливается *Proxy,* опять же неважно каким клиентом;

Если же связь изнутри к *Proxy*-машине есть и эта *Proxy*-машина может общаться с внешним миром, можно переходить к настройке *squid*. *Squid* надо устанавливать из *packages* или *ports*, тогда есть уверенность, что все его компоненты будут размещены по нужным директориям. После установки должно получиться примерно следующее:

- двоичные файлы должны находиться в каталоге */usr/locl/sbin*;
- в каталоге */usr/local/etc* должна появиться директория *squid*, в которой лежит *squid.conf* – это его конфигурационный файл, его надо будет редактировать;
- в каталоге */usr/local/etc/rc.d* появился файл *squid.sh* – это основной стартовый файл (дело в том, что система при старте просматривает этот каталог и все, что найдет там типа **.sh,* запустит автоматически). Но можно его запустить и вручную, просто написав *./squid.sh.* Если такого файла нет, то при перезагрузке системы *squid* не будет запускаться;
- в каталоге */usr/local* образовалась директория *squid*, в которой две поддиректории: *cache* – там будет его кэш и *logs* – логи. Там же должен быть файл (возможно он появится после первого запуска Proxy-сервера) *squid.out* –

это основной лог-файл, в котором и будут сообщения об ошибках, если *squid* почему-либо не сможет стартовать нормально.

Для начала *squid.conf* можно редактировать незначительно. Там стоят значения «по умолчанию» и они вполне приемлемы. Единственное, что необходимо сделать – определиться, сколько мегабайт диска следует выделить под кэш. По умолчанию – 100 Мб. Для изменения этого значения найдите в *squid.conf* строчку

```
cache_swap 100
```

и поставьте подходящее значение. Максимальный объем можно оценить исходя из пропускной способности канала Internet за одни сутки.

Скорее всего, понадобится еще одно исправление. Дело в том, что нельзя запускать Proxy-сервер от имени *root*. Поскольку при старте машины *squid.sh* будет исполняться от имени *root*, то надо сделать следующее. Надо найти в конфигурационном файле строчку

```
cache_effective_usernobodynogroup
```

и «раскомментировать» ее. Это будет означать, что в процессе работы *squid* будет иметь права «псевдоюзера» *nobody*.

На самом деле это не совсем правильно. Правильнее завести нового «псевдоюзера» *squid* (или *www, cache и т.д.*) и в конфигурационном файле вписать именно его, а не *nobody*.

После этого надо будет изменить владельца для директории */usr/local/squid*. Для этого выполните команду

```
chown -R nobody /usr/local/squid
```

Здесь **-R** означает, что меняется владелец не только директории, но и рекурсивно меняется владелец всего содержимого поддиректорий. Если Вы завели специального «псевдоюзера», то, естественно, в команде вместо *nobody* укажите его имя. Теперь осталось сформировать «внутреннюю структуру кэша». Кстати, если этого не сделать, то *squid* при запуске сам подскажет вам: «запустите программу squid –z». Естественно, это надо сделать. При том следует учесть, что */usr/local/sbin* обычно не прописан в PATH даже у пользователя *root*, поэтому лучше набрать полный путь:

```
/usr/local/sbin/squid–z
```

Теперь можно попытаться запустить *squid*. Зайдите в */usr/local/etc/rc.d* и запустите *./squid.sh*. На консоли должно появиться сообщение от *squid*: "Ready to serve requests" («готов обслуживать запросы»).

Настройка Proxy-сервера

Общая настройка Proxy-сервера чаще всего не вызывает сложностей. Сложности обычно вызывают три обстоятельства: настройка *ACL* (*access*

control list – список прав доступа) и правил для них, настройка дополнительных программ вроде «баннерорезалок» и настройка ограничений использования канала.

Настройка ACL

Обратимся к соответствующему месту файла squid.conf и прокомментируем его.

ACL прописываются в виде строки *acl имя_acl тип_aclпараметры ACL или acl имя_acl тип_acl «файл»* – при этом в файле сохраняется по одному значению на строку.

Итак, сначала типы списков.

acl aclname src ip-address/netmask

в этом *acl* описывается *ip*-адрес или сеть, принадлежащая клиентам *squid*. Например:

acl vasya src 192.168.1.1/255.255.255.255

описывает единственную машину с адресом 192.168.1.1 и назначает ей ACL с именем *vasya*.

acl office src 192.168.1.1/255.255.255.0

описывает диапазон машин с адресами 192.168.1.1-.254 и назначает этому ACL имя *office*. Если диапазон необходимо сузить, то необходимо либо изменить маску подсети, либо воспользоваться явным указанием: *acl vip_user src 192.168.1.1-192.168.1.5/255.255.255.0.* Здесь *squid* выбирает тот диапазон адресов, который окажется меньше либо по маске, либо по явному указанию.

acl aclname dst ip-address/netmask

этот тип ACL описывает уже сервер, страницы с которого будут запрашивать клиенты. Следует отметить, что в этом типе ACL задается не символьный адрес сервера, а *ip*.

acl aclname srcdomain.domain.ru

описывает клиентов, но уже не по *ip*-адресам, а по реверсным DNS. Это значит, что нет разницы, какие *ip*-адреса принадлежат клиентам, главное, чтобы они определялись *dns*. Соответственно под это правило попадут все клиенты, стоящие в домене *domain.ru*.

acl aclname dstdomain .domain.ru

описывает сервер. Сравнивается с запросом из URL. Под этот ACL попадут все серверы третьего уровня домена *domain.ru*.

acl aclname srcdom_regex [-i] xxx acl aclname dstdom_regex [-i] xxx

– описания аналогичны предыдущим, но теперь для выяснения, подходит ли правило под запрос, используются *regex*-правила. Если символьный адрес не смог определиться из *ip*-адреса, к запросу будет применена строка *none*.

acl aclname time [day-abbrevs] [h1:m1-h2:m2]

- ACL, описывающий время. Коды дней недели определяются так: *S* – Sunday – Воскресенье, *M* – Monday – Понедельник, *T* – Tuesday – Вторник, *W* – Wednesday – Среда, *H* – Thursday – Четверг, *F* – Friday – Пятница, *A* – Saturday – Суббота;

- вместо *h1:m1* и *h2:m2* вносится время. Следует запомнить – *h1:m1* всегда должно быть меньше *h2:m2*.

Итак, *acl worktime time MTWHF 08:00-17:00* описывает рабочее время с понедельника по пятницу, с 8 утра до 5 вечера, *acl weekday time SA* описывает целиком субботу с воскресеньем, а *acl evening time 17:00-23:59* описывает время до полуночи. Если необходимо описать всю ночь, то приходится заводить два ACL: первый – с вечера до полуночи, а второй – с полуночи до утра.

acl aclname url_regex [-i] ^http://

- *regex*-правила, применяемые ко всему URL.

acl aclname urlpath_regex [-i] \.gif$

- аналогичные правила, применяемые к URL.

acl aclname port 80 70 21

- ACL, описывающий порты. Вместо простого перечисления можно указать диапазон, например, 1-1024.

acl aclname proto HTTP FTP

- ACL, описывающий протокол, по которому клиент желает сделать запрос на сервер.

acl aclname method GET POST

- метод, которым передаются данные клиента серверу.

acl aclname browser [-i] regexp

- *regexp*-запрос на клиентский браузер. Вычисления основаны на заголовке *User-Agent*, который пересылает браузер.

acl aclname ident username

- ACL описывает имя пользователя, от которого запущена программа на клиентской машине. Имя узнается с помощью *ident*-сервера.

acl aclname ident_regex [-i] pattern

то же самое, но основанное на *regex*-правилах.

acl aclname proxy_auth username acl aclname proxy_auth_regex [-i] pattern

- ACL, описывающий имя пользователя. Это имя возвращает внешняя авторизующая программа.

acl aclname maxconn number

- это правило сработает, если клиент сделает больше *number* запросов к кешу.

acl req_mime_type mime-type1

- правило, срабатывающее при *upload* файлов клиентом. Следует подчеркнуть – *uplode*, а не скачивание.

Выше представлены не все описания ACL, но большинство, необходимых в повседневной практике. Для более полного знакомства с описанием следует обратиться к исходному тексту файла *squid.conf* .

Итак, создадим правила обычной сети:

aclallsrc 0.0.0.0/0.0.0.0 aclofficesrc 192.168.1.0/255.255.255.0 all

-правило, описывающее все машины, и *office* – описывающее все машины в подсети 192.168.1.0.

http_access allow office http_access deny all

Эти два правила описывают полный доступ машинам, описываемым *acl office*, и запрещают доступ машинам, описываемым *all*. В приведенном примере есть конфликт в описания прав доступа: машины, попадающие под правило *all* (а по этому правилу все запрещено) не могут использовать Proxy-сервер. Тут в дело вступает порядок просмотра ACL – они просматриваются в порядке объявления, и если сработало одно правило, то другие уже не просматриваются.

К примеру, если мы введем в дополнение ACL

acl vasya src 192.168.1.100/255.255.255.255

и расположим правила так:

http_access allow office http_access deny vasya http_access deny all ,

то машина с *ip*-адресом 192.168.1.100 по-прежнему будет иметь возможность соединяться через Proxy-сервер;

а если так:

http_access deny vasya http_access allow office http_access deny all ,

то все будет в порядке. Остальные офисные машины не попадают под действие первого правила.

Если в списке нет ни одного правила, то запрос будет отвергнут. Если ни одно правило не сработало, то за основу берется последнее. Если, к примеру, заменить предпоследнее правило на *http_access allow all*, то нашим Proxy-сервером смогут пользоваться абсолютно все (кроме *vasya*), кто сможет соединиться с портом *squid*. Так что будьте внимательнее. Разработчики *squid* предусмотрели вариант, даже если последнее правило будет разрешающим для всех, то запрос будет отвергнут. Это поможет избежать дыр в Proxy-сервере.

На основе этих же списков-правил так же управляется и доступ к другим возможностям Proxy-сервера (см. файл *squid.conf*, где все расписано).

Предположим, что в сети появились пользователи, которые честно подключаются к серверу и начинают выкачивать гигабайтами запрещенную информацию. При этом занесение этих сайтов в *deny*-список вызывает их возмущение.

На этот счет придумали много вещей, но самым эффективным остается сокращение канала для таких пользователей: доступ есть, но качается плохо, возразить им нечего – такая ситуация в Internet не редкость.

Итак, давайте разберемся с «траффик-шейпингом» – именно так это называется. В *squid* же это называется *delay-pool*. Заметим, что *squid* при сборке должен быть собран с опцией *--enable-delay-pools*.

Сначала разберемся, какие есть пулы. Пулы делятся на три класса. Первый, и самый простой, это когда всему *ACL* ограничивается трафик до определенной величины. Второй – когда отдельно ограничивается трафик для одной машины из подсети и для всей подсети. И третий класс – когда ограничивается трафик для отдельных машин, для подсети класса *C* или меньше и для подсети класса *B*.

Давайте рассмотрим ситуацию, когда в сети завелся «дискокачальщик».

delay_pools

1 – у нас всего один пул.

delay_class 1 1

первый пул первого класса.

delay_access 1 allow vasya

delay_access 1 deny all

В первый пул попадают только машины, описываемые ACL *vasya*. Остальные работают, как им положено, ведь им доступ к первому пулу запрещен.

delay_parameters 1 800/64000

Теперь файлы и страницы объемом до 64 Кб будут скачиваться на максимальной скорости, а то, что больше этого – на скорости 800 б/с.

Или совсем уж радикальная мера:

delay_parameters 1 800/800

и «злобный качальщик» все будет качать на скорости 800 б/с.

Но даже в не очень большой сети будут возникать ситуации, когда все хотят что-то качать, в итоге никому ничего не хватает.

Исправляем строчку с *delay_pools* на *delay_pools 2.* Теперь у нас будет два пула.

delay_class 2 2

второй пул будет второго класса (совпадение номеров чисто случайно) – первый – это *vasya.*

delay_access 2 allow office

delay_access 2 deny all

Во второй пул попадают только машины с ACL *office*.

delay_parameters 2 64000/64000 4000/4000

В итоге вся подсеть, описываемая *office*, будет использовать канал не больше 512 Кбит/с (64 Кб/с), но каждый отдельный хост будет качать не более 4 Кб/с. Этим правилом очень легко разграничить по скорости разные подсети, использующие один канал.

К примеру, у нас есть две подсети, описываемые *office* и *office1*. При этом *office* не должна иметь никаких ограничений на канал (примем канал за 256 Кбит) в целом, но каждый из *office* не должен качать быстрее 6 Кб/с. А *office1* – это пользователи, которым всем и 5 Кб/с хватит.

Создаем два пула второго класса и прописываем для них ACL. Затем определяем этим пулам параметры.

delay_parameters 3 -1/-1 6000/6000

это определение для *office* (ему отдан номер пула 3).

delay_parameters 4 5000/5000 -1/1

а это для *office1*.

В итоге после применения этих правил получаем все, что заказано – первый офис грузит канал как хочет (-1/-1), но никто из сотрудников больше 6 Кб/с не получает. А второй офис грузит канал не больше 5 Кб/с, но в распределении этих 5 Кб/с между сотрудниками нет никаких правил.

Понятно, что в описание пулов можно заложить и другие параметры, например, время, место доступа и т. д. Остается еще одна маленькая вещь, которую нельзя оставить без внимания. И эта вещь – навязанная реклама через баннеры и другие объекты. Для того, чтобы такую рекламу не пропустить на браузер, каждый URL, который передается *squid*, первоначально передается *редиректору*. И тот либо возвращает прежний URL в случае, если все в порядке, либо возвращает тот, который, по его мнению, более правильный. Возможна ситуация перехвата обращения к баннерам и счетчикам и вместо них подгрузить любой файл. В итоге страницы можно заполнить прозрачными окошками.

Итак, в *squid.conf* прописываем строку:

redirect_program /squid/bin/redirector ,

где */squid/bin/redirector* – путь до выполняемой программы, которая как раз и обеспечивает разбор URL. Ее можно написать на чем угодно, но наиболее предпочтительным является язык Perl – так как именно он предназначен для подобного рода работ. Полная версия *редиректора* расположена по адресу *http://linuxnews.ru/redirector*.

Описание директив squid

Описание директив содержится непосредственно в файле конфигурации *squid.conf* и в документации, прилагаемой к данной лабораторной работе.

3.3.3. Задания на выполнение практической работы

1. Сконфигурировать Proxy-сервер на порт 3128; разрешить доступ с *ip* 10.10.146.150 с 10-00 до 15-00 ч; запретить доступ с *ip* 10.10.146.176 с 10-00 до 15-00 ч; запретить доступ к доменам **.khb.ru*.
2. Сконфигурировать Proxy-сервер на порт 8080; разрешить доступ с *ip* 10.10.146.176 с 10-00 до 15-00 ч; запретить доступ к файлам **.gif*; запретить доступ к домену **.khb.ruc**ip*10.10.146.150.
3. Сконфигурировать Proxy-сервер на порт 12345; разрешить доступ с *ip* 10.10.146.176; запретить доступ к файлам **.cgi*; запретить доступ к домену **.khb.ruc**ip* 10.10.146.150 с 10-00до 15-00 ч.
4. Сконфигурировать Proxy-сервер на порт 1345; запретить доступ с *ip* 10.10.146.176; запретить метод *POST*; запретить доступ к домену **.khstu.ru*и к файлам **.gif*c*ip* 10.10.146.150 с 10-00 до 15-00 ч.

3.3.4. Контрольные вопросы

1. Назначение Proxy-сервера.
2. В каком файле содержится информация о конфигурации Proxy-сервера?
3. Какие протоколы кэшируются Proxy-сервером?
4. Какие условия должны быть выполнены перед установкой Proxy-сервера?
5. Что такое список прав доступа?
6. Общий синтаксис ACL.

3.4. VPN. СОЗДАНИЕ ВИРТУАЛЬНОЙ ЧАСТНОЙ СЕТИ НА БАЗЕ PPTP СЕРВЕРА POPTOP

3.4.1. Цель занятия

Установка и конфигурирование сервера POPTOP под ОС Linux. Конфигурирование клиентов виртуальной частной сети под ОС Linux и ОС Windows.

3.4.2. Методические указания по теме

Общие 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 Protocolover 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.

Порядок установки и конфигурирования сервера виртуальной частной сети:

1. Установить необходимые пакеты 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 необязательным. 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 Linuz (apt, 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 серверов-зеркал, хранящих разные версии пакетов.

1.1. Загрузите пакеты

Пакеты, необходимые для практической работы, можно загрузить с адреса ftp://kid/pub/LECTURES/5KURS/ProectKSM-labs/lab1-vpn/, а именно ppp-2.4.4.tar.gzипptpd-1.3.3.tar.gz.

1.2. Разархивируйте пакеты

Разархивирование можно сделать командами:

```
$ tar -zvxf ppp-2.4.4.tar.gz
$ tar -zvxfpptpd-1.3.3.tar.gz
```

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

1.3. Установите пакет ppp

Войдите в корневую папку пакета ppp-2.4.4. Соберите пакет ppp, это можно сделать следующими командами:

```
$ ./configure
```

По умолчанию пути дальнейшей установки файлов пакета настроены на /usr/local (бинарные файлы) и /etc (настройки). Их можно поменять параметрами --prefix и --sysconfdir.

```
$ make
```

Следующие команды обычно необходимо выполнять с правами администратора:

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

1.4. Установите пакет pptpd

Войдите в корневую папку пакета pptpd-1.3.3. Соберите пакет pptpd, это можно сделать следующими командами:

```
$ ./configure
```

По умолчанию путь дальнейшей установки файлов пакета настроены на /usr/local. Его можно поменять параметром --prefix. Также существуют параметры для более детального задания путей для каждой части пакета, их все можно увидеть, набрав команду ./configure --help.

```
$ make
```

Следующие команды обычно необходимо выполнять с правами администратора:

```
# make install
```

2. Создать конфигурационные файлы и скрипт запуска сервера PPTP

Примеры конфигурационных файлов пакетов ppp и pptpd находятся соответственно в папках ./ppp-2.4.4/scripts и ./pptpd-1.3.3/samples. Описание параметров конфигурационных и командных файлов описаны в соответствующих man-страницах.

PPTP сервер POPTOP можно запустить следующей командой:

```
# ./pptpd \
--conf <путь к конфигурационному файлу pptpd>/pptpd.conf
```

По умолчанию полный путь к конфигурационному файлу pptpd: /etc/pptpd.conf. При выполнении данной команды может возникнуть необходимость загрузки модулей ядра, обеспечивающих работы ppp и pptpd, это может быть выполнено следующим командами:

```
# modprobe ipip
# modprobe ip_gre
```

и т.д., загружая необходимые модули ядра. Пример конфигурационного файла pptpd.conf:

```
################################################################################
# $Id: pptpd.conf,v 1.10 2006/09/04 23:30:57 quozl Exp $
#
# Пример конфигурационного файла Poptop /etc/pptpd.conf
#
# Изменения вступают в силу после перезапуска демона pptpd.
################################################################################

# TAG: ppp
# Путь к pppd, по умолчанию '/usr/sbin/pppd'
#
ppp /usr/local/sbin/pppd

# TAG: option
# Указывает местонахождения файла опций PPP, так называемого peer-a.
# По умолчанию PPP читает '/etc/ppp/options'
#
option /etc/ppp/options.pptpd

# TAG: debug
# Включает отладочный вывод в syslog.
#
debug

# TAG: stimeout
# Указывает таймаут (в секундах) для старта управляющего соединения.
#
# stimeout 10
```

```
# TAG: noipparam
# Запрещает передачу IP клиента в PPP,
# что в противном случае делается по умолчанию.
#
#noipparam

# TAG: logwtmp
# Использовать wtmp(5) для записи клиентских подключений и отключений.
#
#logwtmp

# TAG: bcrelay <if>
# Включает режим броадкастового релея к клиентам с указанного интерфейса <if>
#
#bcrelay eth1

# TAG: delegate
# Делегировать распределения клиентских IP-адресов pppd.
#
# Без данной опции, по умолчанию, pptpd управляет списком
# клиентских IP-адресов и передаёт следующий свободный адрес pppd.
# С данной опцией, pptpd не передаёт IP-адрес, и следовательно pppd может
# использовать radius или файл chap-secrets для выделения адресов.
#
#delegate

# TAG: connections
# Ограничивает количество допустимых подключаемых клиентов.
#
# Если pptpd распределяет IP-адреса (то есть опция delegate не используется),
# тогда количество подключений также ограничивается опцией remoteip.
# По умолчанию 100.
#connections 100

# TAG: localip
# TAG: remoteip
# Устанавливают диапазон локальных и удалённых IP-адресов.
#
# Данные опции игнорируются в случае использования опции delegate.
#
# Любые адреса работают на протяжении периода, пока локальная станция
# производит их маршрутизацию. Если вы хотите использовать MS-Windows сеть,
# вы должны
# использовать IP-адреса, не входящие в диапазон адресов локальной сети (LAN),
# и использовать опцию proxyarp в файле опций pppd, или запустить bcrelay.
#
# Вы можете задавать отдельные IP-адреса, разделяемые запятой, или вы можете
# указать диапазон, или и то и другое вместе, например:
#
# 192.168.0.234,192.168.0.245-249,192.168.0.254
#
# ВАЖНЫЕ ОГРАНИЧЕНИЯ:
```

```
#
# 1.Между запятыми и в адресах не допускаются пробелы.
#
# 2. Если вы задали IP-адресов больше, чем значение опции connections,
#   распределение начнётся с начала списка  и будет выбирать последовательно
#   IP- адрес, пока не достигнет значения connections.
#   В противном случае будет проигнорировано.
#
# 3. Сокращения не допустимы!
#   то есть 234-8 не означает диапазон от 234 до 238,
#   необходимо назначать 234-238, если вы имеете это в виду.
#
# 4. Если вы задали один локальный IP, это нормально – он будет назначен
#   всем локальным IP. Вы ДОЛЖНЫ задать, по крайней мере, один удалённый IP
#   для каждого клиента, которые будут работать одновременно.
#
# (Рекомендуемо)
localip 192.168.0.100
remoteip 192.168.0.200-250
# или
#localip 192.168.0.234-238,192.168.0.245
#remoteip 192.168.1.234-238,192.168.1.245
```

Пример конфигурационного файла options.pptpd:

```
################################################################################################
# $Id: options.pptpd,v 1.11 2005/12/29 01:21:09 quozl Exp $
#
# Пример файла опций Poptop PPP /etc/ppp/options.pptpd
# Опции используются PPP, когда клиент совершает подключение.
# Этот файл указывается опцией option в /etc/pptpd.conf.
# После его изменения и сохранения, изменения будет применены
# к следующим подключениям. См. "man pppd".
#
# Подразумевается изменение данного файла для задания параметров,
# необходимых вашей системе.
# Для настроек по умолчанию необходим PPP 2.4.2 и модуль ядра MPPE.
################################################################################

# Аутентификация (идентификация)

# Имя локальной системы для процедуры аутентификации.
# (должно совпадать со вторым полем строк в /etc/ppp/chap-secrets)
name pptpd

# Вырезать префикс домена из имени пользователя перед аутентификацией.
# (работает, если вы используете pppd с патчем chapms-strip-domain)
#chapms-strip-domain

# Шифрование
# (Ниже перечислены разные версии PPP поддержкой шифрования,
# выберите, какую из секций вы будете использовать).
```

```
# ppp-2.4.2 под BSD лицензией интегрирован с MPPE, модуль ядра ppp_mppe.o
# {{{
refuse-pap
refuse-chap
refuse-mschap
# Требуется аутентификации клиента с использованием MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] аутентификации.
require-mschap-v2
# Требуется MPPE 128-bit шифрование
# (заметьте, что MPPE требует использования MSCHAP-V2 при аутентификации)
require-mppe-128
# }}}

# ppp-2.4.1 под OpenSSL лицензией работает с MPPE через внешние интерфейсы,
# модуль ядра mppe.o
# {{{
#-chap
#-chapms
# Требуется аутентификации клиента с использованием MS-CHAPv2 [Microsoft
# Challenge Handshake Authentication Protocol, Version 2] аутентификации.
#+chapms-v2
# Требуется MPPE шифрование
# (заметьте, что MPPE требует использования MSCHAP-V2 при аутентификации)
#mppe-40 # должна быть использована одна из опций или 40-bit, или 128-bit
#mppe-128
#mppe-stateless# }}}

# Сеть и маршрутизация
# Если pppd выступает в роли сервера для Microsoft Windows клиентов, данная
# опция позволяет pppd сообщать один или два DNS (DomainNameServer)
# адреса клиентам. Первое значение данной опции
# задаёт первичный DNS адрес; второе значения (если задано)
# задаёт вторичный DNS адрес.
#ms-dns 10.0.0.1
#ms-dns 10.0.0.2

# Если pppd выступает в роли сервера для MicrosoftWindows или "Samba"
# клиентов, данная опция позволяет pppd сообщать один или два адреса
# WINS (Windows Internet Name Services) сервера клиентам. Первое
# значение данной опции задаёт первичный WINS адрес; второе значение
# (если задано) задаёт вторичный WINS адрес.
#ms-wins 10.0.0.3
#ms-wins 10.0.0.4

# Добавить значение в ARP [Address Resolution Protocol] таблицу
# данной системы с IP-адресом PPTP соединения и Ethernet адресом данной системы.
# Это позволит сделать PPTP соединение доступным для других
# систем в локальной сети ethernet.
# (вам это не нужно, если ваш PPTP сервер отвечает за маршрутизацию
# пакетов клиентам -- James Cameron)
proxyarp
```

```
# Обычно pptpd передаёт IP-адрес pppd, но если pptpd была установлена
# опция delegate в pptpd.conf или параметр --delegate в командной строке,
# тогда pppd будет использовать chap-secrets или radius для определения
# IP-адреса клиента. Локальный IP-адрес, используемый на сервере,
# обычно такой же, как адрес сервера. Для того, чтобы его переопределить,
# укажите здесь необходимый IP-адрес.
# (вы не должны использовать это, если вы не используете опцию delegate)
#10.8.0.100

# Журналирование (логирование)

# Включить вывод отладочной информации.
# (смотрите настройки демона syslog куда pppd посылает отладочную информацию)
#debug

# Выводить все настроечные значения, которые установлены.
# (обычно запрашивается подтверждения данной опции через список рассылки)
dump

# Дополнительные опции
# Создавать лок-файл UUCP-стиля для псевдо-tty для обеспечения исключающего доступа.
lock
# Выключить BSD-Compress компрессию
nobsdcomp
# Выключить Van Jacobson компрессию
# (необходимо в некоторых сетях с Windows 9x/ME/XP клиентами, см. сообщение в
# poptop-server от 14th Апреля 2005 от Pawel Pokrywka,
# http://marc.theaimsgroup.com/?t=111343175400006&r=1&w=2)
novj
novjccomp

# выключить логирование в stderr, далее это может быть переправлено pptpd,
# который может быть возвращён обратно
nologfd

# здесь помещаются используемые дополнительные модули (плагины)
# (помещение их выше может вызвать посылку сообщения на pty)
```

3. Тестирование работы сервера PPTP

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

```
$ telnet<IP-адрес сервера PPTP> 1723
```

При установленных в конфигурационных файлах опции debug, работы сервера детально журналируются демоном syslog, по умолчанию сообщения сервера будут помещаться в файл /etc/log/messages. Динамически просматривать изменение данного файла можно запустив следующую команду:

```
# tail -d /etc/log/messages
```

4. Добавить 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.

Рассмотрим способ добавления пользователя через конфигурационный файл.

Пример конфигурационного файла chap-secrets:

```
# Данные для аутентификации, использующие CHAP
# клиент сервер пароль IP-адрес
username pptpd password *
```

5. Настроить клиента созданной виртуальной частной сети из ОС Windows

Для того, чтобы клиент, работающий на ОС Windows, мог подключаться к серверу PPTP, в составе ОС должен присутствовать Microsoft VPN Adapter.

Рассмотрим пример подключения ОС Windows XP, в ней данный драйвер установлен по умолчанию. Для создания VPN-подключения необходимо выполнить следующие шаги:

1. Пуск->Панель управления->Сетевые подключения->Создание нового подключения.
2. Будет запущен мастер создания подключения.
3. На втором шаге выбрать «Подключить к сети на рабочем месте».
4. На следующем шаге «Подключение к виртуальной частной сети».
5. Далее указать название подключения «test».
6. Далее «Не набирать номер для предварительного подключения».
7. Далее указать IP-адрес запущенного PPTP-сервер.
8. Готово.

После запуска созданного подключения будет затребован логин и пароль пользователя, далее необходимо ввести данные, введённые на 4-м шаге. В случае изменения конфигурационного файла options.pptpd может возникнуть необходимость редактирования свойств подключения.

6. Настроить клиента созданной виртуальной частной сети из ОС Linux

Подключение ОС Linux к серверу PPTP может осуществляться аналогично ОС Windows с применением разнообразных визардов как графических (pptpconfig и др.), так и текстовых (pptp-command и др.), существующих в современных дистрибутивах. Однако для примера рассмотрим механизм подключения ОС Linux посредством ручного редактирования конфигурационных файлов и запуска демона pppd. Для подключения клиента должен быть установлен пакет pptp (PPTP-driver

http://pptpclient.sourceforge.net/). В случае его отсутствия необходимо произвести его установку одним из методов, описанных в пункте 1.

Подключение может быть выполнено следующей командой:

pppdcallvpn
(Опция nodetach полезна для отладки),

где vpn – имя файла с настройками подключения, находящегося в папке/etc/ppp/peers/.

Пример конфигурационного файла vpn (параметры аналогичны конфигурационному файлу options.pptpd):

```
name username
remotename vpn
ipparam vpn
#Использовать программу pptp как псевдотерминал для pppd
pty "pptp IP.IP.IP.IP  --nolaunchpppd"
connect /bin/true
defaultroute
refuse-eap
refuse-chap
refuse-mschap
require-mschap-v2
require-mppe-128
noauth
lock
```

Пример конфигурационного файла chap-secrets:

```
username * password * IP.IP.IP.IP
```

7. Включить маршрутизацию пакетов в ОС Linux

По умолчанию возможность маршрутизации 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

8. Создать частную приватную сеть (VPN)

Сеть состоит из двух рабочих станций:

−WS1-настроенный сервер PPTP под ОС Linux;

−WS2- настроенные клиенты сервера PPTP под ОС Windows и Linux.

Сетевой интерфейс WS1 должен иметь IP-адрес, маршрутизируемый в сети университета, и IP-адрес (на том же интерфейсе – алиас) тестовой сети, немаршрутизируемой в сети университета.

Сетевой интерфейс WS2 должен иметь IP-адрес тестовой сети.

После установления VPN-подключения WS2 должна получить IP- адрес из сети университета, т.е. получив доступ к её ресурсам.

3.4.3. Задания на выполнение практической работы

1) Настроить PPTP-сервер в среде Linux.
2) Настроить клиент PPTP-сервера в среде Linux и проверить его работу.
3) Настроить клиент PPTP-сервера в среде Windows и проверить его работу.

3.4.4. Контрольные вопросы

1. Причины использования VP-сетей?
2. Перечислите наиболее распространённые протоколы идентификации в PPP? В чем их отличие?
3. Какие пакеты необходимо установить в систему для возможности создания PPTP-сервера? Перечислите их основные составляющие?
4. В чём разница между PPP и PPTP?
5. Как добавить пользователя на сервере?
6. Как выполняется настройка клиента из ОС Windows/Linux?
7. Перечислите основные параметры конфигурационного файла PPTP.

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Олифер Н.А., Олифер. Компьютерные сети. – СПб.: Питер, 2011.
2. Таненбаум Э. Компьютерные сети. – 4-е изд.– СПб.: Питер, 2007.

СОДЕРЖАНИЕ