

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ
Черниговский государственный технологический университет

СИСТЕМНЫЕ СЕРВИСЫ TCP/IP СЕТЕЙ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторному практикуму по дисциплине

"Компьютерные сети"

для студентов направления 0915 - "Компьютерная инженерия"

Чернигов ЧДТУ 2008

ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ.....	3
ВВЕДЕНИЕ.....	5
<u>1 ЛАБОРАТОРНАЯ РАБОТА № 1. КОНФИГУРИРОВАНИЕ СЛУЖБЫ ИМЁН DNS В КОРПОРАТИВНОЙ СЕТИ.....</u>	<u>6</u>
<u>1.1 Цель работы.....</u>	<u>6</u>
<u>1.2 Краткие теоретические сведения.....</u>	<u>6</u>
1.2.2 Записи ресурсов в базе данных домена.....	9
1.2.3 Прямая зона DNS.....	9
1.2.4 Обратная зона DNS.....	12
1.2.5 Особенности размещения и конфигурирования серверов для корпоративной сети.....	14
1.2.6 Установка ПО сервера DNS.....	14
1.2.7 Конфигурирование сервера DNS.....	15
<u>1.3 Тестирование работы сервера имён.....</u>	<u>18</u>
<u>1.4 Ход работы.....</u>	<u>20</u>
<u>1.5 Содержание отчета.....</u>	<u>20</u>
<u>1.6 Контрольные вопросы.....</u>	<u>20</u>
<u>2 ЛАБОРАТОРНАЯ РАБОТА № 2. СЕРВИСЫ ДЛЯ СЕТЕЙ WINDOWS. НАСТРОЙКА СЕРВИСА SAMBA.....</u>	<u>21</u>
<u>2.1 Цель работы.....</u>	<u>21</u>
<u>2.2 Краткие теоретические сведения.....</u>	<u>21</u>
2.2.1 Обзор протоколов.....	22
2.2.2 Описание компонент пакета Samba.....	26
2.2.3 Конфигурационный файл "/etc/smb.conf".....	28
<u>2.3 Ход работы.....</u>	<u>33</u>
2.3.1 Установка простого сервиса разделения ресурсов.....	33
2.3.2 Тестирование сервера.....	35
<u>2.4 Содержание отчета.....</u>	<u>36</u>
<u>2.5 Контрольные вопросы.....</u>	<u>36</u>
<u>3 ЛАБОРАТОРНАЯ РАБОТА № 3. КОНФИГУРИРОВАНИЕ СЛУЖБЫ ДНСР В КОРПОРАТИВНОЙ СЕТИ.....</u>	<u>37</u>
<u>3.1 Цель работы.....</u>	<u>37</u>
<u>3.2 Краткие теоретические сведения.....</u>	<u>37</u>
3.2.1 Общие сведения.....	37
3.2.2 ДНСР сервер под Unix.....	38
3.2.3 ДНСР сервер под Window.....	40
3.2.4 ДНСР-клиент под Unix.....	41
3.2.5 ДНСР-клиент под Windows.....	42

<u>3.3</u> <u>ХОД РАБОТЫ</u>	<u>42</u>
<u>3.4</u> <u>СОДЕРЖАНИЕ ОТЧЕТА</u>	<u>43</u>
<u>3.5</u> <u>КОНТРОЛЬНЫЕ ВОПРОСЫ</u>	<u>43</u>
ЛИТЕРАТУРА	44

ВВЕДЕНИЕ

Межсетевой протокол IP на сегодняшний день доминируют как в локальных, так и в глобальных сетях. С развитием сети Интернет стек протоколов TCP/IP “оброс” огромным количеством сетевых сервисов, что и обусловило доминирование данного семейства протоколов во всех разновидностях сетей. Однако существует ряд сервисов, без которых полноценное функционирование корпоративной сети или невозможно, или крайне затруднительно. К таким сервисам следует отнести службу доменных имён DNS, службу конфигурирования сетевых интерфейсов DHCP, ряд служб для рабочих станций под управлением ОС Windows, основанных на семействе протоколов SMB.

В данном учебном пособии приводятся краткие теоретические сведения, необходимые для понимания работы указанных сервисов, рассматриваются основные конфигурационные параметры и варианты настроек. Также приводятся пошаговые инструкции по настройке конкретного ПО, реализующего данные сервисы.

Следует отметить, что данное пособие ориентировано исключительно на свободное программное обеспечение и не рекламирует каких бы то ни было коммерческих производителей ПО. Выполнение работы в данном пособии ориентировано на дистрибутив Fedora Linux 9, однако практически никаких отличий не будет для дистрибутивов RedHat, CentOS и подобных им, использующих менеджер пакетов *rpm* и *yum*.

Для других POSIX-совместимых ОС, использующих иные менеджеры пакетов, необходимо ознакомиться с руководством по утилитам управления пакетами, в остальном различия не существенны.

Коммерческие ОС, в частности, ОС Windows применяются только в случае, если предоставляемые сервисы ориентированы на эти ОС.

Цикл лабораторных работ, предлагаемый в данном пособии, поможет студентам научиться конфигурировать базовые системные сервисы TCP/IP сетей, такие как сервис имен DNS и сервис динамического конфигурирования хостов DHCP, а также получить базовые навыки в настройке системного сервиса Samba.

1 Лабораторная работа № 1. Конфигурирование службы имён DNS в корпоративной сети

1.1 Цель работы

Цель данной работы – научиться конфигурировать и тестировать службу имен для корпоративной сети на основе сервера DNS bind.

1.2 Краткие теоретические сведения

1.2.1.1 Служба имен в Интернете

Служба доменных имён DNS является важнейшей системной службой в TCP/IP сетях. Назначение DNS – преобразование символьных имен в IP адреса и наоборот, а так же предоставление дополнительной информации о хостах и группах хостов (доменах), такой как адреса почтовых обменников, публичные ключи служб и т.п. В сети Интернет служба DNS оперирует распределённой иерархической базой данных в виде дерева имен, находящейся на множестве различных хостов. Ответственность за корректность базы данных в каждом узле дерева возлагается на администратора соответствующего домена.

В сети Интернет корнем дерева является домен “.”. Полное - абсолютное или полностью определенное, fully qualified domain name - доменное имя заканчивается точкой, обозначающей корень доменного дерева, но часто эта завершающая точка опускается. Анализ имени производится справа налево. Самая правая секция имени характеризует страну (для каждой страны мира выделен свой домен с двух символьным именем в соответствии со стандартом ISO, например, ua – Украина, ru – Россия, uk – Англия и т.п.) или характер организации (образовательная, коммерческая, правительственная и т.п.). Таим образом, домены верхнего уровня (Top Level Domains, TLD) - это хорошо известные всем домены, такие как org (бесприбыльная организация), com (коммерческая организация), gov (государственное учреждение), mil (военное предприятие или организация), edu (учебное заведение), int (международная организация), net (большая сеть) и т.д. Ответственность за TLD лежит на международной неприбыльной организации Internet Corporation for Assigned Names and Numbers (ICANN, www.icann.org).

Ниже по дереву имен расположены домены второго и последующих уровней. В некоторых странах существуют домены, подобные исторически первым TLD, например, org.ua, gov.ua, edu.ua и т.п. В Украине для каждой области выделены двухбуквенные домены, такие как sp.ua – Чернигов, dp.ua – Днепропетровск, и т.д., хотя исторически сложившиеся домены, такие как kiev.ua, kharkov.ua, chernigov.ua, тоже поддерживаются. Маленький фрагмент

интернетовской иерархии имен показан на рисунке 1. Число уровней реально больше, но обычно не превышает 5.

Таким образом, в службе DNS каждый сервер отвечает за определенную зону (зона ответственности) - т.е. свою часть дерева доменных имен, хранит соответствующие базы данных и отвечает на запросы. При этом вышестоящие по дереву серверы имеют информацию об адресах нижестоящих серверов, что обеспечивает связность дерева. Говорят, что вышестоящий сервер делегирует нижестоящему серверу полномочия по обслуживанию определенной зоны. Важно понимать различие между доменом и зоной. Домен - это поддерево дерева доменных имен. Зона - это часть дерева, за которую отвечает тот или иной DNS-сервер.

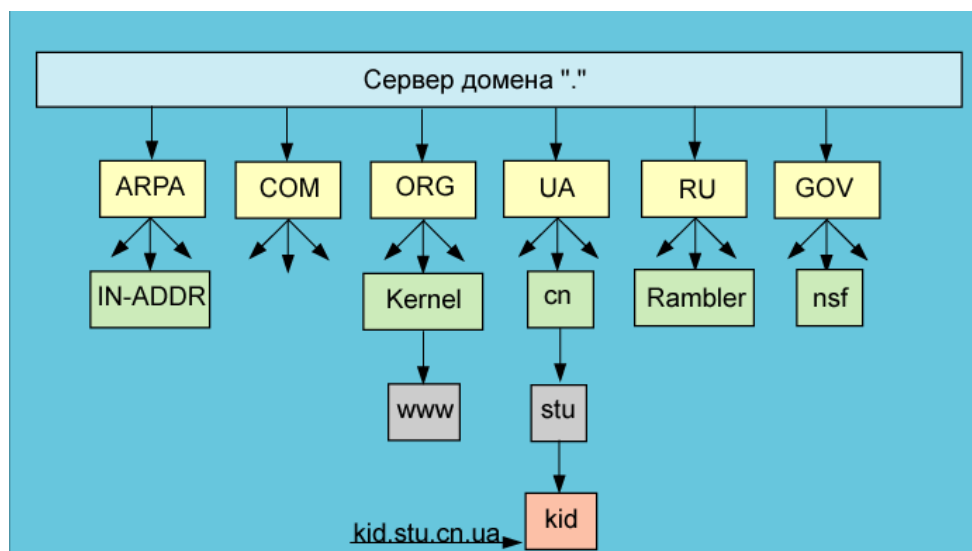


Рисунок 1 – Фрагмент иерархии имен в Интернет

За каждую зону DNS отвечает не менее двух серверов. Один из них является первичным, primary, или, в новой терминологии — master. остальные - вторичными, secondary, или slave. Первичный сервер содержит оригинальные файлы с базой данных DNS для своей зоны. Вторичные серверы получают эти данные по сети от первичного сервера и периодически запрашивают первичный сервер на предмет обновления данных. Признаком обновления данных служит увеличение серийного номера в записи SOA – см ниже. В случае, если данные на первичном сервере обновлены, вторичный сервер запрашивает "передачу зоны" ("zone transfer")- т.е. базы данных требуемой зоны. Передача зоны происходит с помощью протокола TCP, порт 53, в отличие от запросов, которые направляются на UDP/53.

Изменения в базу данных DNS могут быть внесены только на первичном сервере. С точки зрения обслуживания клиентских запросов первичные и вторичные серверы идентичны. Рекомендуется, чтобы первичный и вторичные серверы находились в разных сетях - для увеличения надежности обработки запросов на случай, если сеть одного из

серверов становится недоступной. Серверы DNS не обязаны находиться в том домене, за который они отвечают.

Вторичный сервер необязательно получает данные непосредственно с первичного сервера; источником данных может служить и другой вторичный сервер. В любом случае сервер-источник данных для данного вторичного сервера называется "главным" ("master").

У каждого домена есть свои "писанные правила", именуемые "полиси", в соответствии с которыми можно получить имя хоста или субдомена (чаще говорят "зоны") в данном домене. Самым простым примером таких правил могут служить полиси домена com. Если данное имя не зарегистрировано или не находится в процессе регистрации, то по уплате определенной суммы любое физическое или юридическое лицо может получить это имя от регистратора домена com.

Процесс получения собственного домена называется "делегирование", что, собственно, и отражает суть действий: субдомен передается в полное управление его администратору. Для получения субдомена в каком-то домене необходимо в соответствии с полиси обратиться к регистратору, выполнить формальные процедуры, а так же ряд технических действий по настройке сервера DNS, которые, собственно и являются предметом данной лабораторной работы.

Различают 3 режима работы сервера DNS:

master (primary). Данный режим используется администратором зоны, файлы баз данных ведутся вручную на этом сервере. Данный сервер является абсолютным авторитетным источником информации для данной зоны;

slave (secondary). Данный режим используется по просьбе администратора зоны, которая автоматически регулярно копируется с master сервера. Данный сервер является авторитетным источником информации для данной зоны;

hint (caching). Режим кэширования всех запросов, попадающих в определённую зону, обычно ".", т.е. кэшируются все запросы. Такой сервер обычно используется для ускорения работы с сетью.

Для каждой зоны, обслуживаемой данным сервером, может быть выбран тот или иной режим. Обычно для зоны "." все сервера конфигурируются по типу hint, что позволяет кэшировать все запросы пользовательских рабочих станций на время жизни конкретной записи DNS. Это значительно ускоряет обработку локальных запросов.

База данных DNS для каждого домена в простейшем случае представляет собой набор текстовых файлов, которые системный администратор ведет на главном сервере имен этого домена. В этих файлах содержатся директивы синтаксического анализатора (\$ORIGIN, \$TTL) и записи о ресурсах.

1.2.2 Записи ресурсов в базе данных домена

Файл любой зоны начинается с записи Start Of Authority, **SOA**. Эта запись является заголовочной и содержит информацию о размещении зоны, о почтовом адресе ответственного лица и о базовых временных параметрах записей данной зоны.

Файл прямой зоны содержит стандартные записи ресурсов базы данных DNS для преобразования доменных имен хостов в данной зоне в IP-адреса, определения авторитарных DNS-серверов данной зоны, определения хостов-обработчиков почты для доменных имен в данной зоне и др.

Файлы баз данных DNS состоят из стандартных записей ресурсов. В общем виде стандартная запись ресурса связывает данные определенного типа с некоторым именем и формируется по шаблону:

имя [время_жизни_записи] IN тип_записи данные

Именем является некоторое доменное имя (необязательно имя физически существующих хоста или домена). Если поле "имя" пусто, то значение этого поля берется из предыдущей записи. Данными может быть, например, IP-адрес хоста, если имя относится к хосту, или DNS-сервер домена, если имя относится к домену, и т.п.

Время жизни записи определяет время хранения информации этой записи в кэше запросившего запись сервера в секундах и указывается, только если оно отличается от времени жизни, определенного для всей зоны в записи SOA.

Основные типы записей:

SOA (Start Of Authority) - заголовок зоны;

NS (Name Server) - сервер DNS;

A (Address) - IP-адрес для хоста;

MX (Mail Exchanger) - почтовый обменник;

CNAME (Canonical Name) - каноническое имя, псевдоним хоста;

PTR (Pointer) - указатель по обратной зоне, фактически — имя хоста;

1.2.3 Прямая зона DNS.

Рассмотрим примеры файлов базы данных DNS. Первой рассмотрим прямую зону для приватной части корпоративной сети, домен "stu.", файл db.stu.

```
$ORIGIN .
stu 28800 IN SOA ns.stu. dnsmaster.stu. (
    2005033100 ; Serial
    28800 ; Refresh
    7200 ; Retry
    604800 ; Expire
    86400 ; Time To Live)
```

```

; authoritative name servers for zone
28800 IN NS ns.stu.
28800 IN NS ns1.stu.
; mail exchangers for entire zone
28800 IN MX 10 stalker.stu.
28800 IN MX 20 cs.stu.
$ORIGIN stu.
; name servers glue records
ns IN A 192.168.0.10
ns1 IN A 192.168.0.14
;servers
dragon IN A 192.168.0.17
auth IN CNAME dragon.stu.
cs IN A 192.168.0.14
stalker IN A 192.168.0.10
www IN CNAME stalker.stu.
mail IN CNAME stalker.stu.
ftp IN CNAME stalker.stu.
www.docs IN CNAME cs.stu.
kid IN A 192.168.0.12
; workstations
ie-21-7 IN A 192.168.3.40
ie-21-8 IN A 192.168.3.41
ie-21-9 IN A 192.168.3.42
vc-105-1 IN A 192.168.66.2

```

Первая строка – это макрос, говорящий, что все имена далее следуют непосредственно за доменом “точка”. Таким образом, для приватной сети мы используем имена в нашем приватном дереве относительно нашего собственного корня “.”. Следует помнить, что для сервера, разрешающего одновременно и имена в корпоративной сети, и имена в интернете, имя зоны следует выбирать из 3-х символов, не совпадающих с именами TLD.

Первой записью всегда идет **SOA** (Start of Authority), в которой указывается имя зоны (“stu.”, или макрос @), TTL, т.е. время жизни этой записи, дальше – ключевые слова IN (Internet records) и **SOA**. Далее идут параметры зоны: имя основного сервера DNS, почтовый адрес администратора зоны, однако вместо символа “@” там стоит точка, поскольку @ - это ссылка на имя зоны. Сразу за открывающейся скобкой находится серийный номер данного файла, обычно в формате гтггммддNN. Серийный номер необходимо увеличивать при каждом изменении файла, что бы ведомые сервера идентифицировали изменения и обновили файлы баз данных с главного сервера. Далее следуют стандартные времена в секундах для данной зоны:

Refresh – время, по истечении которого вторичные сервера должны обновить данные с первичных серверов (zone transfer);

Retry – время, через которое вторичные сервера должны совершить повторную попытку обновления, если предыдущая попытка не удалась;

Expire – время, через которое вторичные сервера должны выбросить запись о зоне и считать ее недоступной, если обновления не удались.

TTL – стандартное время жизни записей из данной зоны для кеширующих серверов.

Следующая группа записей является так же обязательной и указывает на авторитетные сервера имен для данной зоны - записи типа *NS*. Авторитетным является сервер, на котором информация соответствует реальному состоянию зоны, т.е. регулярно обновляется (см. выше). Крайне желательно, чтобы имена, указанные в этой секции, имели соответствующие адресные *IN A* записи в этой же базе данных.

Ниже следует секция почтовых обменников, т.е. записи типа *MX* (Mail Exchanger). Они указывают на сервера электронной почты, способные принимать почту для всего домена по протоколу SMTP. Чем меньше цифра перед именем, тем больший приоритет имеет данный почтовый сервер. Как правило, запись с наивысшим приоритетом относится к серверу, на котором почта заканчивает свой путь, а другие записи относятся к серверам-релеям, на которых почта может сохраняться некоторое время, пока основной почтовый сервер для зоны не доступен. Естественно, записи *MX* на релеи нельзя расставлять произвольно, поскольку релей обязательно должен быть сконфигурирован для приёма почты данного домена. При отсутствии записи *MX* для какого-либо доменного имени, почта, адресованная с этим доменным именем, будет доставляться непосредственно на хост, имеющий такое имя. Однако, такого хоста может не быть, в этом случае почта вернется отправителю с сообщением об ошибке.

Ниже, после макроса “\$ORIGIN stu.”, задающего суффикс для всех записей ниже, следуют записи типа *IN A*, предназначенные для задания соответствия между именем хоста в зоне и его IP адресом.

Для задания псевдонимов хостам используется запись *CNAME* (Canonical Name). Псевдонимы удобны для указания на стандартные сервисы, такие как www, mail, ftp, а так же для задания псевдонимов, используемых для создания виртуальных серверов (см. www, docs).

Рассмотрим теперь файл зоны stu.cn.ua. Данная зона мало чем отличается от предыдущей зоны внешне.

```
$ORIGIN .
stu.cn.ua 28800 IN SOA ns.stu.cn.ua. nsmaster.stu.cn.ua (
    2005033100 ; Serial
    28800 ; Refresh
    7200 ; Retry
    604800 ; Expire
    86400 ; Time To Live
)
; authoritative name servers for zone
IN NS ns.stu.cn.ua.
```

```

IN NS ns1.stu.cn.ua.
IN NS ns.cn.ua.
; mail exchangers for entire zone
IN MX 10 stalker.stu.cn.ua.
IN MX 15 cs.stu.cn.ua.
IN MX 20 relay1.cn.ua
; name servers glue records
ns.cn.ua IN A 212.86.96.10
$ORIGIN stu.cn.ua.
ns IN A 195.69.76.130
ns1 IN A 195.69.76.134
;servers
dragon IN A 195.69.76.137
auth IN CNAME dragon.stu.cn.ua.
cs IN A 195.69.76.134
stalker IN A 195.69.76.130
www IN CNAME stalker.stu.cn.ua.
mail IN CNAME stalker.stu.cn.ua.
ftp IN CNAME stalker.stu.cn.ua.
www.docs IN CNAME cs.stu.cn.ua.
; workstations
admin IN A 195.69.76.139

```

Конфигурация данной зоны практически повторяет предыдущую зону, однако отличие в том, что данная зона является субдоменом домена cn.ua. Значит, она должна быть делегирована в соответствующей зоне cn.ua примерно так, как показано в следующем фрагменте:

```

$ORIGIN cn.ua.
stu IN NS ns.stu.cn.ua.
IN NS ns1.stu.cn.ua.
IN NS ns.cn.ua.
$ORIGIN .
ns.stu.cn.ua IN A 195.69.76.130
ns1.stu.cn.ua IN A 195.69.76.134

```

Как видно из приведенного фрагмента, в “материнской” зоне cn.ua находятся только записи о серверах имен для делегированной зоны stu.cn.ua. Управление остальной информационной базой зоны передается на сервера ns.stu.cn.ua и ns1.stu.cn.ua.

1.2.4 Обратная зона DNS

Теперь рассмотрим файлы обратных зон, предназначенные для проведения обратного DNS-преобразования, т.е. "IP-адрес в доменное имя".

Для частных блоков адресов, таких как 10.0.0.0/8, 172.16.0.0/12 и 192.168.0.0/16 никаких проблем с делегированием, в общем-то, нет,

поскольку эти адреса не маршрутизируются в Интернете и нужны только внутри приватной сети.

```
$ORIGIN .
0.168.192.IN-ADDR.ARPA. 86400 IN SOA ns.stu. dnsmaster.stu. (
    2006060200
    86400
    14400
    3600000
    345600
)
86400 IN NS ns.stu.
86400 IN NS ns1.stu.
$ORIGIN 0.168.192.IN-ADDR.ARPA.
10 IN PTR stalker.stu.
14 IN PTR cs.stu.
17 IN PTR dragon.stu.
12 IN PTR kid.stu.
```

Стоит обратить внимание, что имя зоны состоит из развёрнутых по отношению к записи адреса цифр. Для адресного блока 192.168.0.0/24 имя зоны 0.168.192.IN-ADDR.ARPA. INADDR.ARPA. - это специальный домен верхнего уровня, отведенный для делегирования обратных зон. В файле обратной зоны присутствует, конечно же, запись **SOA**, как минимум пара записей типа **NS** об официальных авторитетных серверах и записи типа **PTR** (Pointer), ставящие в соответствие адреса и имена. Обратная зона для публичных адресов 195.69.76.0/24 приведена ниже:

```
$ORIGIN .
76.69.195.IN-ADDR.ARPA. 86400 IN SOA ns.stu.cn.ua
dnsmaster.stu.cn.ua (2004060200 86400 14400 3600000 345600 )
IN NS ns.stu.cn.ua.
IN NS ns1.stu.cn.ua.
$ORIGIN 0.168.192.IN-ADDR.ARPA.
10 IN PTR stalker.stu.cn.ua.
14 IN PTR cs.stu.cn.ua.
17 IN PTR dragon.stu.cn.ua.
12 IN PTR kid.stu.cn.ua.
```

Отличие данной зоны от предыдущей опять же только в том, что она является публичной и должна делегироваться в соответствии с правилами выдачи и регистрации IP адресов. Вкратце необходимо отметить следующее. Выдача блоков адресов потребителям производится локальными Интернет регистратурами (LIR), которые, в свою очередь, получают их от региональных регистратур. В Европе это – бесприбыльная организация RIPE (<http://www.ripe.net/>), финансируемая провайдерами. Основные функции региональных регистратур – координация использования IP адресов и,

соответственно, маршрутизации в регионе. Для получения своего блока публичных адресов необходимо заполнить соответствующие формы RIPE и направить их вашему LIR.

1.2.5 Особенности размещения и конфигурирования серверов для корпоративной сети

Поскольку сервис DNS является критическим для функционирования сети, то в сети должно быть как минимум 2 сервера. Обычно размещают их так, как показано на рисунке 2. Для внутренней сети доступны оба сервера, а для внешней – только внешний сервер. Внутренний сервер является первичным для внутренних доменов и использует внешний в качестве форварда (forwarding server), поскольку напрямую не видит домен “.”. Внешний сервер в общем случае не должен отвечать на рекурсивные запросы извне, поскольку он может быть использован как платформа для DDoS атак на другие сервера.

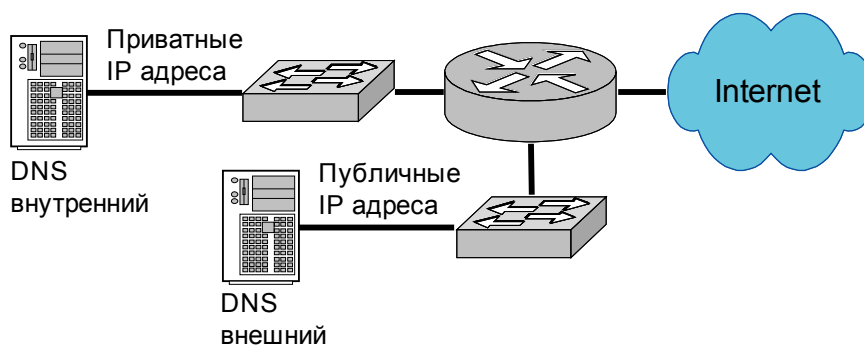


Рисунок 2 – Расположение серверов имен в сети

1.2.6 Установка ПО сервера DNS

Наиболее популярным ПО, реализующим сервис DNS, является сервер *bind* (Berkeley Internet Name Daemon), поддерживаемый Internet Software Consortium (<http://www.isc.org>). Существуют лёгкие реализации, предназначенные для работы в качестве кеширующих серверов, есть реализации, специфичные для конкретных коммерческих ОС, однако на практике лучше использовать *bind*, поскольку он распространяется под лицензией GPL, т.е. вместе с исходным кодом, что гарантирует отсутствие давно не исправляемых уязвимостей и прочих неприятностей закрытого кода.

Пакет *bind* поставляется практически со всеми дистрибутивами Linux и xBSD.

Для работы сервера в ОС Fedora Linux не обходимо установить следующие пакеты: *bind-utils* – утилиты для работы с DNS и тестирования сервера, *bind* – собственно сервер, *bind-chroot* – файлы, необходимые для запуска сервера в индивидуальном окружении в режиме *chroot*. Запуск сервера

в этом режиме минимизирует потери при взломе системы через работающий сервис.

Проверить, установлены ли пакеты, можно командой

```
rpm -qa | grep bind
```

Если пакеты не установлены, установите их командой

```
yum install bind-utils bind bind-chroot
```

При установке сервера автоматически создается конфигурация для кеширующего сервера зоны «.» и для первичных серверов локальных зон.

Для небольшого сервера на несколько мелких зон конфигурации зон обычно хранятся в текстовых файлах. В нашем случае – это `/var/named/chroot/var/named/*`. Для больших зон, содержащих миллионы записей, используются специальные модули хранения, которые могут сохранять данные зон в реляционных БД (PostgreSQL) либо на сервере LDAP.

1.2.7 Конфигурирование сервера DNS

Рассмотрим подробнее пример файла конфигурации для такого случая: наш сервер является основным для внутренней корпоративной сети 192.168.0.0 и внутреннего домена “stu.”, и кеширующим для домена “.”. Все пути, приведенные ниже, в случае работы сервера в окружении chroot, нужно понищать как относительные к `/var/named/chroot`. Файл конфигурации `/etc/named.conf` приведен ниже:

```
options {
    directory "/etc/namedb";
    forward first;
    forwarders {
        195.69.76.130;
    };
};

// caching server for root domain
zone "." {
    type hint;
    file "named.root";
};

// it's not necessary but good to resolve localhost
zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "localhost.rev";
};
```

```

};
// private zone for our network
zone "stu" {
    type master;
    file "db.stu.private";
    allow-transfer{
        195.69.76.130;
    };
    allow-query{
        192.168.0.0/16;
    };
};
// Inverse zone for private zone stu
zone "0.168.192.IN-ADDR.ARPA" {
    type master;
    file "db.192.168.0";
    allow-transfer{
        195.69.76.130;
    };
    allow-query{
        192.168.0.0/16;
    };
};
};

```

В приведенном выше файле конфигурации все интуитивно понятно, однако стоит особо отметить, что сервер в DNS с адресом 195.69.76.130 используется как форвард и как вторичный сервер для внутренних доменов. Предложение *allow-transfer* используется для ограничения полной перекачки зоны только на вторичный сервер. Предложение *allow-query* указывает серверу, что отвечать на запросы по данным зонам можно только хостам из приведенного блока адресов.

Ниже приведена конфигурация второго сервера. Кроме того, он является первичным для публичных зон `stu.cn.ua` и `76.69.195.IN-ADDR.ARPA`.

```

options {
    directory "/etc/namedb";
    forward first;
    forwarders {
        212.86.96.10;
    };
};

// caching server for root domain
zone "." {
    type hint;
    file "named.root";
};

```

```

// it's not necessary but good to resolve localhost
zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "localhost.rev";
};
// private zone for our network
zone "stu" {
    type slave;
    file "db.stu.private";
    master {
        192.168.0.10;
    };
    allow-query{
        192.168.0.0;
    };
};
// Inverse zone for private zone stu
zone "0.168.192.IN-ADDR.ARPA" {
    type slave;
    file "db.192.168.0";
    masters {
        192.168.0.10;
    };
    allow-query{
        192.168.0.0;
    };
};

// public zone for our network
zone "stu.cn.ua." {
    type master;
    file "db.stu.cn.ua";
    allow-transfer{
        212.86.96.10;
    };
};
// Inverse zone for public zone stu.cn.ua.
zone "76.69.195.IN-ADDR.ARPA" {
    type master;
    file "db.195.69.76";
    allow-transfer{
        212.86.96.10.
    };
};
};

```

Заметим, что здесь появилось предложение *allow-query*, необходимое для того, чтобы данный сервер мог отвечать на запросы о внутренних зонах только во внутреннюю сеть. Предложение *allow-transfer* используется так же, как и в предыдущем случае и ограничивает полную перекачку всей зоны

только серверу с адресом 212.86.96.10, который является вторичным для данной зоны. Этот же сервер используется и как форвард. Строго говоря, такой необходимости нет, но в ситуации, когда почему-то пол-Интернета не видно, а этот форвард находится в пределах досягаемости и видит весь Интернет, использование форварда оказывается оправданным.

Естественно, многие вопросы остались за пределами этого короткого обзора. Например, конфигурирование мощной системы логов сервера bind, специальные ресурсные записи и т.д.

1.3 Тестирование работы сервера имён

Тестирование работы сервера производится командой dig, которая позволяет производить произвольные запросы к указанному в командной строке серверу. Ниже приведен пример запроса к DNS серверу, расположенному на локальном компьютере на предмет записи SOA для зоны stu.

```
al@stalker$>dig @127.0.0.1 SOA stu
; <<>> DiG 8.3 <<>> @127.0.0.1 SOA stu
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12648
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2,
;; ADDITIONAL: 2
;; QUERY SECTION:
;; stu, type = SOA, class = IN
;; ANSWER SECTION:
stu. 8H IN SOA ns.stu. dnsmaster.stu. (
2005033100 ; serial
8H ; refresh
2H ; retry
1W ; expiry
1D ) ; minimum
;; AUTHORITY SECTION:
stu. 8H IN NS ns.stu.
stu. 8H IN NS ns1.stu.
;; ADDITIONAL SECTION:
ns.stu. 8H IN A 192.168.0.10
ns1.stu. 8H IN A 192.168.0.14
;; Total query time: 19 msec
;; FROM: stalker.stu.cn.ua to SERVER: 127.0.0.1
;; WHEN: Wed Sep 14 14:37:22 2005
;; MSG SIZE sent: 21 rcvd: 134
```

Разберём вывод команды. Следует отметить, что вывод содержит собственно ответ на заданный вопрос и дополнительные сведения, отмеченные как комментарии знаками “;” в начале строки. Поскольку мы

запрашивали запись SOA, то она и показана в секции ответов. Далее следует секция авторитетности, в которой указываются авторитетные сервера для данной зоны, и, наконец, дополнительная секция, где обычно указываются IP адреса (записи IN A) для авторитетных серверов данной зоны. Ниже приведен еще один пример запроса типа A.

```
al@stalker$dig @127.0.0.1 A stalker.stu
; <<>> DiG 8.3 <<>> @127.0.0.1 A stalker.stu
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14417
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2,
;; ADDITIONAL: 2
;; QUERY SECTION:
;; stalker.stu, type = A, class = IN
;; ANSWER SECTION:
stalker.stu. 8H IN A 192.168.0.10
;; AUTHORITY SECTION:
stu. 8H IN NS ns.stu.
stu. 8H IN NS ns1.stu.
;; ADDITIONAL SECTION:
ns.stu. 8H IN A 192.168.0.10
ns1.stu. 8H IN A 192.168.0.14
;; Total query time: 3 msec
;; FROM: stalker.stu.cn.ua to SERVER: 127.0.0.1
;; WHEN: Wed Sep 14 14:47:33 2005
;; MSG SIZE sent: 29 rcvd: 112
```

Как видно из примера, кроме затребованной записи A, выданы опять же две секции об авторитетных серверах, т.е. тех серверах, где информация о данной зоне наиболее достоверна.

Если запрос попадает к кэширующему серверу, то информация о зоне в нём может быть устаревшей. Запросы для проверки обратной зоны нужно давать в форме полного имени записи PTR. См. пример ниже:

```
alukin@stalker$dig @127.0.0.1 SOA 1.1.168.192.IN-ADDR.ARPA
; <<>> DiG 8.3 <<>> @127.0.0.1 SOA 1.1.168.192.IN-ADDR.ARPA
; (1 server found)
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 22408
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1,
;; ADDITIONAL: 0
;; QUERY SECTION:
;; 1.1.168.192.IN-ADDR.ARPA, type = SOA, class = IN
;; AUTHORITY SECTION:
1.168.192.IN-ADDR.ARPA. 4D IN SOA ns.stu. dnsmaster.stu. (
2001101800 ; serial
```

```
1D ; refresh
4H ; retry
5w6d16h ; expiry
4D ) ; minimum
;; Total query time: 8 msec
;; FROM: stalker.stu.cn.ua to SERVER: 127.0.0.1
;; WHEN: Wed Sep 14 14:54:58 2005
;; MSG SIZE sent: 42 rcvd: 94
```

1.4 Ход работы

Выполнение данной лабораторной работы состоит из следующих шагов:

1. Создайте файлы зон для прямой и обратной приватной зоны. Возьмите блок адресов 172.16.X.0, где X – номер машины в классе. Сконфигурируйте первичный сервер DNS для этих зон на локальном компьютере и запустите его. Не забудьте, что все сообщения выводятся не на консоль, а в системный журнал. Сообщения удобнее всего просматривать в отдельном терминале командой:

```
tail -f /var/log/messages
```

2. Проведите тестирование созданных прямой и обратной зон при помощи команды **dig**.

3. Проведите тестирование разрешения внешних имён вашим сервером. Запросите, например информацию о зоне slashdot.org.

4. Сконфигурируйте ваш сервер как вторичный для зон, которые создал ваш сосед по лаборатории. Произведите корректировку и проверку записей зон на предмет записей типа NS. Произведите проверку командой **dig**, обращая особое внимание на секцию “AUTHORITY SECTION”.

1.5 Содержание отчета

Отчет должен содержать файлы зон, файл конфигурации сервера и результаты проверок. Наличие комментариев и выводов необходимо.

1.6 Контрольные вопросы

Почему для корпоративных зон удобнее использовать 3-х буквенные имена?

В каком случае сервер имён считается авторитетным?

Какие параметры отвечают за время обновления зоны вторичным сервером?

Как обеспечить защиту зоны от скачивания и от просмотра?

Какая запись RR применяется при создании виртуальных серверов?

Какая запись RR задаёт почтовый обменник для всей зоны?

Какой файл содержит адреса корневых серверов имен, необходимых для инициализации кеша и рекурсивных запросов?

Что такое рекурсивный запрос?

Как производится установка ведомого сервера для конкретной зоны?

2 Лабораторная работа № 2. Сервисы для сетей Windows. Настройка сервиса Samba

Если Интернет хоть что-то и доказал, то только то, что огромное количество приматов, стучащее достаточно долгий период времени по клавиатуре, может и на самом деле производит удивительно полезное ПО. С другой стороны, если вы соберете некоторое количество этих приматов вместе, разместите их в кабинках и научите выполнять цирковые трюки ...

... ладно, мы сейчас потратили кучу сил, расчищая тот кавардак, что они натворили в этих кабинках.

Cristopher R. Hertel, «Implementing CIFS».

2.1 Цель работы

Цель данной работы – приобретение навыков в настройке сервиса на основе свободно распространяемого пакета Samba, обеспечивающего основные сервисы для рабочих станций под управлением ОС Windows.

2.2 Краткие теоретические сведения

В любой корпоративной сети, независимо от ее размера, необходимо выделять общие ресурсы, такие как принтеры, диски, службы времени, службы авторизации и аутентификации, репозитории ПО и т.д. В операционных системах семейства Windows для этих целей используются протоколы Server Message Block (SMB) и, позже - Common Internet File System (CIFS). Эти протоколы являются развитием спецификаций NetBIOS и NetBEUI, разработанных в середине 80-х компаниями IBM и Microsoft для работы в локальных сетях. В POSIX-совместимых ОС данное семейство протоколов реализовано в пакете свободного ПО Samba (<http://www.samba.org>).

В виду закрытости протоколов SMB и CIFS в начале разработка Samba велась методом реверс-инжиниринга, однако в 2004-м году Еврокомиссия вынудила Microsoft смягчить политику лицензирования интеллектуальной собственности и проект Samba в 2007-м году получил доступ к документации по указанным протоколам, что дало возможность более точно и надежно реализовать сервисы указанных протоколов. На момент написания этой книги свободная реализация SMB/CIFS Samba является функционально

полной и предоставляет весь набор сервисов, необходимых для интероперабельности UNIX-подобных ОС и ОС семейства Windows.

2.2.1 Обзор протоколов

Протокол CIFS появился в середине 90-х, когда компания Microsoft подала черновики его спецификаций в организацию Internet Engineering Task Force (IETF). Основная часть спецификации посвящена описанию разделения в сети файловых ресурсов, основанном на уже применяемом, но плохо документированном и уязвимом протоколе SMB.

Протокол SMB был изначально разработан для работы поверх NetBIOS (Network Basic Input Output System) в локальных сетях. Цитируя книгу Кристофера Хертеля «Реализация CIFS», можно сказать, что «NetBIOS - это грязный маленький скелет в шкафу CIFS». До появления Win2K поддержка NetBIOS была необходима для работы SMB. Фактически, «компьютеры» и «службы», которые видны в «Сетевом окружении» - это имена NetBIOS. Для работы не только поверх кадров 802.3, но и поверх высокоуровневого протокола IP, необходимо отображать 16-байтное символьное имя NetBIOS на IP адрес, и механизм такого отображения, а также инкапсуляции протокола NetBIOS в транспортные протоколы TCP и UDP были предложены интернет-сообществом в RFC 1001, RFC1002. С развитием ОС Windows (Windows 3.11) компании Microsoft пришлось реализовать предложенные стандарты, поскольку семейство протоколов TCP/IP стало доминирующим в локальных сетях в виду широкого распространения Интернет. В этой же версии ОС был предложен механизм поиска сервисов, основанный на широковещательных сообщениях о доступных ресурсах. Естественно, такой механизм создавал в сети ситуацию с трафиком, аналогичную шуму на базаре, когда каждый торговец громко рекламирует свой товар. Здесь же была предложена концепция рабочих групп для упрощения управления ресурсами, которая послужила основой для доменов Windows NT.

Поскольку широковещательный механизм анонсов доступных ресурсов не работает за пределами одного сегмента сети, в Windows NT появляется также Windows Internet Name Service (WINS, не путать с DNS), который реализует централизованный механизм анонса и просмотра доступных разделяемых SMB ресурсов. Каждая рабочая станция сообщает серверу WINS о предоставляемых ресурсах и может прочесть у него информацию о доступных ресурсах других рабочих станций. В виду неряшливого проектирования данного сервиса, может пройти около часа, пока ресурсы станут «видны» в сети. Следует также отметить, что назначение имён компьютеров, рабочих групп и ресурсов оставлено за пользователем и практически не контролируется протоколом, что приводит к неизбежным конфликтам и полной неразберихе в сети.

В Win2K наконец-то разработчики компании отделили реализацию SMB от NetBIOS, что и было отражено в поданных в IETF «черновиках»

протокола CIFS, который, кроме базовых сервисов по выделению файловых ресурсов включает также механизмы разрешения имён, авторизации, аутентификации и анонсирования сервисов. Была сделана попытка приблизить эти сервисы к существующим интернет-стандартам, однако закрытые частные модификации стандартных протоколов делают интероперабельность нетривиальной задачей. В реализации Win2K предложено использовать динамическую службу доменных имён — Dynamic DNS (RFC 2136) для именованя компьютеров и рабочих групп, что внесло еще большую путаницу. Кроме того, введен сервис активных каталогов (Active Directory), который является комбинацией протоколов LDAP и Kerberos с модификациями, который заменил традиционную службу просмотра, авторизации и аутентификации. Впрочем, обратная совместимость с более ранними версиями ОС вынудила производителя ПО оставить поддержку служб SMB Windows NT.

Несмотря на столь хаотичное развитие и крайне неудовлетворительную стандартизацию, семейство протоколов SMB/CIFS на сегодня устоялось и широко применяется в корпоративных сетях по вполне понятным причинам.

Рассмотрим более подробно основные механизмы SMB/CIFS в его сегодняшней реализации.

Начнем с именованя ресурсов. Универсальное соглашение об именах UNC — это способ адресации ресурса в сетях CIFS. Например, запись <ftp://kid/home/al/trash.txt> в нотации URL специфицирует следующие элементы: протокол ftp, хост kid, путь /home/al и имя файла trash.txt. В нотации UNC путь выглядит <\\kid\home\al\trash.txt>. Первое, что бросается в глаза — пусть не содержит протокола. Это не потому, что протокол только один, а потому что ОС сама должна как-то выяснить, какой протокол будет использоваться для доступа к данному ресурсу. Забавно? Следующее отличие — в имени хоста. Имя в нотации URL — это всегда имя DNS. В нотации UNC имя проверяется сначала как имя NetBIOS, потом как имя DNS и потом, на всякий случай, как IP адрес. С путем все тоже не совсем здорово. В нотации URL путь к файлу всегда однозначен, будь он относительным или абсолютным. В нотации UNC первый элемент интерпретируется как имя разделяемого ресурса, имя которому дает пользователь и оно в большинстве случаев не связано с путём, по которому ресурс доступен на файловой системе. Это создает проблемы, аналогичные символическим ссылкам в UNIX или ярлыкам в Windows — один и тот же ресурс может быть доступен под разными именами. Последнее, казалось бы уж совсем просто — имя файла — это же просто имя файла! Но и тут есть своя весёлая путаница. Во первых, существуют чувствительные и не чувствительные к регистру файловые системы, во вторых — допустимые и недопустимые символы в именах файлов в различных ОС Windows, в третьих — ограничения на длину имени файла, и, в конце концов, различные кодировки для не ASCII символов. Так что имя файла в нотации UNC может

оказаться совсем не таким, каким оно выглядит на реальной файловой системе. В утешение можно сказать, что всё большую популярность набирает так называемый SMB URL, пример:

<smb://kid:2891/my%2Eresource/my%2Efile.txt>.

Пояснения к данному примеру излишни.

Немного о протоколах. Сервис SMB может работать в двух режимах - в традиционном, с использованием NetBIOS поверх TCP/IP, иногда называемом NBT, и в сыром режиме, то есть без использования АПИ NetBIOS в версиях Windows старше Win2K. При запросе клиента к ресурсу происходит согласование протоколов и его диалектов, а далее используется один из общих для клиента и сервера вариантов. В большинстве случаев — это традиционный NBT, поскольку он поддерживается старшими версиями ОС для совместимости. Останавливаться на протоколах подробно мы не можем в виду ограниченного объёма пособия, однако стоит отметить, что если правильно выделенный, правильно адресованный и авторизованный ресурс оказывается недоступен, значит сервер и клиент не пришли к общему решению в ходе согласования протоколов и вместе работать не могут.

Понятие домена в в SMB/CIFS развивалось от рабочих групп, основанных просто на групповых именах NetBIOS, через концепцию домена Win NT, основанную на сервисе WINS и сервисе просмотра до понятия домена в Win2K, основанного на DDNS и LDAP. В любом случае, под доменом CIFS можно подразумевать некоторое множество хостов, объединенных под общим групповым именем и использующим общий сервис аутентификации и авторизации.

Существует, как всегда в мире Windows, некоторая путаница понятий с доменами. Для прояснения извечного мрака необходимо различать 2 сервиса в домене — сервис просмотра и сервис авторизации.

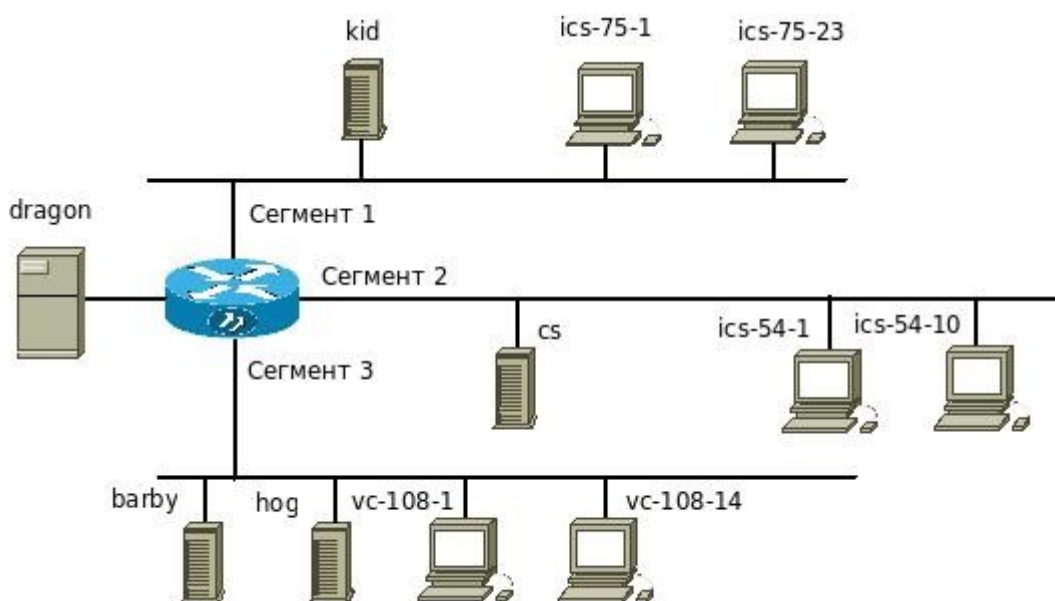


Рисунок 3 – Пример сети из трёх сегментов

Сначала разберём сервис просмотра. На рисунке 3 приведен пример простой сети из 3-х сегментов. В каждом сегменте есть компьютер помощнее, который условно показан как сервер, и несколько рабочих станций. В отдельном сегменте сети, не обязательно сразу же за маршрутизатором, размещен сервер с именем dragon, на котором установлена служба WINS, ну и естественно где-то в сети, хоть и на том же сервере есть служба DNS. На всех остальных компьютерах — рабочие станции с Windows старше NT 4. Для простоты предположим, что у всех компьютеров задана одна и та же рабочая группа STU.

В каждом сегменте, так сказать, «на общественных началах» появится локальный мастер просмотра (Local Master Browser, LMB), потому что после широковещательного обмена пакетами поиска мастера просмотра рано или поздно рабочие станции запустят процедуру выборов главного компьютера, который будет хранить список доступных в сегменте ресурсов и отвечать на широковещательные запросы просмотра. Если какому-то компьютеру администратор (человек) указал работать в качестве предпочтительного мастера просмотра домена (DMB), то он на выборах будет «кричать» дольше всех, и наверное, победит. После победы мастер просмотра будет отвечать на широковещательные запросы и обмениваться с сервером WINS списками доступных ресурсов во всех сегментах, которые работают с данным сервером WINS. Стоит заметить, что адрес сервера WINS явно указывается в настройках сети либо вручную, либо через сервис DHCP. Задача WINS — накапливать информацию о доступных в сети ресурсах и рассказывать о них рабочим станциям, т.е. Ставить в соответствие имена NetBIOS и IP адреса. Кроме того, сервер WINS обычно работает как прокси к DNS и преобразует DNS имена в NetBIOS имена для узлов, у которых NetBIOS имена не указаны явно. Естественно, во избежание путаницы, необходимо давать NetBIOS имена станциям такие же, как и DNS имена.

Другая функция домена — это аутентификация и авторизация. И тут появляется понятие контроллер домена. Его функции обычно совмещены с DMB, что и вносит путаницу. Задача контроллера домена — аутентификация пользователя и авторизация доступа к сетевым ресурсам SMB/CIFS. Различают первичный контроллер домена (PDC) и резервный контроллер (BDC). Когда рабочая группа не является рабочей группой? Когда она является доменом NT.

Аутентификация и авторизация в семействе SMB/CIFS развивалась от простейшей реализации с пересылкой пароля в открытом виде в первых вариантах протокола, через шифрование по алгоритму DES, хеширование по MD4 до использования механизма сеансовых «билетов» по протоколу Kerberos в последних реализациях CIFS и Active Directory.

Следует различать 3 схемы авторизации ресурсов: авторизация по ресурсу, авторизация по пользователю и авторизация по домену. Авторизация по ресурсу подразумевает один пароль для ресурса, общий для всех пользователей. Авторизация по пользователю подразумевает индивидуальный пароль для пользователя ко всем ресурсам данной рабочей станции, возможно, с исключениями доступа к отдельным ресурсам. Авторизация по домену подразумевает логин и пароль для пользователя, либо другие средства аутентификации, которые открывают доступ ко всем ресурсам домена, выделяемым различными рабочими станциями и серверами в домене, с соответствующим данному пользователю и группе правами.

Наиболее используемой является схема авторизации по пользователю, хотя правильнее было бы использовать доменную схему.

2.2.2 Описание компонент пакета Samba

Пакет Samba позволяет практически в полной мере заменить файловый сервер Windows, включая не только выделение файловых ресурсов и принтеров, но и сервис просмотра, авторизации, аутентификации, управление пользователями и доменами, сервис Active Directory. Samba распространяется свободно под лицензией GPL v.3 и поставляется проактически сов семи свободными ОС. Здесь рассматривается стабильная на сегодня версия 3.

Список выполняемых файлов Samba можно получить командой:

```
$ rpm -ql `rpm -qa | grep samba` | grep bin/
```

Подробно ознакомиться с каждым можно, прочитав соответствующие разделы документации, выполнив команду:

```
$man samba
```

Остановимся на самых важных и наиболее часто используемых компонентах.

Основные серверные компоненты:

/usr/sbin/nmbd - сервер преобразования имен и адресов NetBIOS, необходим для нормальной работы сервиса просмотра;

/usr/sbin/smbd - основной сервер — файловый сервис, сервис печати, сервис времени и т.д.;

/usr/sbin/winbindd - сервер импорта пользователей и групп с PDC для локальной аутентификации;

/usr/sbin/swat - средство конфигурирования Samba с web-интерфейсом, обычно запускается как сервис через демон **xinetd**;

/etc/init.d/smb, /etc/init.d/nmb, /etc/init.d/winbind - управляющие скрипты инициализации соответствующих сервисов. Именно эти скрипты должны использоваться для запуска и останова сервисов.

Следует отметить, что у скрипта **/etc/init.d/smb** есть два режима рестарта – **restart** и **reload**, которые радикально отличаются следующими особенностями:

restart перезапускает процессы **smbd** и **nmbd** со сбросом текущих соединений. Как правило, клиенты сами производят автоматическое восстановление соединения с ресурсами, однако если в момент перезапуска были открыты файлы, то возможны проблемы с клиентскими приложениями (например, MS Office и 1С);

reload заставляет **smbd** и **nmbd** только перечитывать файлы конфигурации без перезапуска и сброса соединений. При этом старые соединения продолжают существовать по старым правилам, а ко всем новым соединениям будут применены уже новые правила на основании файлов конфигурации.

Клиентские компоненты:

/usr/bin/smbclient - интерактивное приложение для просмотра сетевых ресурсов;

/sbin/mount.smb, /sbin/mount.smbfs, usr/bin/smbumount, /usr/bin/smbmount, и т.п. - средства монтирования/размонтирования сетевых файловых систем.

Утилиты:

/usr/bin/smbpasswd, /usr/bin/pdbedit - управление пользователями и подключением к домену;

/usr/bin/wbinfo - отображение списка пользователей, импортированных **winbindd**;

/usr/bin/testparm - проверка синтаксиса конфигурационных файлов;

/usr/bin/smbstatus - отображение статуса процессов **smbd** и **nmbd**;

/usr/bin/nmblookup - программа разрешения имен WINS (аналог **nslookup** для DNS).

Все файлы конфигурации и авторизации Samba расположены в каталоге **/etc/samba** и его подкаталогах. Рассмотрим их подробнее:

lmhosts – то же, что и **/etc/hosts**, но предназначен для преобразования IP в NetBIOS. Как правило, содержит только одну запись: 127.0.0.1 localhost.

smb.conf – основной конфигурационный файл Samba. Он нужен не только серверной части, но и всем остальным компонентам этой системы;

smbpasswd – аналог **/etc/passwd** и **/etc/shadow** — файл пользователей сервера Samba с паролями. С точки зрения безопасности имеет ту же ценность, что и **/etc/shadow** — а потому права доступа должны быть root.root 0600. Соответствие пользователей Samba и системных производится на основе общего UID; данный файл используется Samba при отсутствии данных о пользователе на PDC или при отсутствии самого PDC;

smbusers – файл соответствий имен сетевых пользователей SMB и локальных пользователей; это удобный метод для организации административных и гостевых входов на сервер. Соответствие пользователей Samba и системных производится на основе символьных имен;

/var/log/samba/* – лог-файлы серверной части Samba. Из них log.smbd, log.nmbd, log.winbind — журналы соответствующих процессов, а все прочие — логи взаимодействия сервера с отдельными клиентскими хостами в формате именования по умолчанию **log.<Client_NetBIOS_NAME>**. При превышении заданного в **smb.conf** предела производится ротация логов и формируются файлы ***.old**;

/var/cache/samba/* – файлы (как правило, двоичные базы данных), формируемые в процессе работы различных компонентов Samba. Наиболее примечательны текстовые файлы browse.dat и wins.dat.

/var/lib/samba/* – служебные файлы сервера.

2.2.3 Конфигурационный файл "/etc/smb.conf"

Файл **/etc/smb.conf** - это основной конфигурационный файл сервера Samba, в котором вы можете определить каталоги, к которым предоставляете доступ, с каких IP адресов разрешен доступ и пр. Первые несколько строк в секции [global] содержат глобальные конфигурационные директивы, которые являются общими для всех разделяемых ресурсов (пока они не переписаны в конкретных секциях для каждого ресурса), далее идут секции, отвечающие за конкретные ресурсы. Существует множество опций, и нужно обязательно прочитать документацию, поставляемую вместе с Samba, чтобы получить информацию о каждой из них.

Следующий пример представляет собой минимальную рабочую конфигурацию для сервера Samba, работающего в режиме контроллера домена. Основные параметры, связанные с безопасностью, выделением ресурсов и оптимизацией работы прокомментированы ниже, остальные –

оставлены на самостоятельное рассмотрение и изучение. Для получения подробной информации не обходимо дать команду ***man smb.conf***.

```
[global]
dos charset = CP1251
unix charset = UTF-8
workgroup = ics-73
netbiosname = ics-73-10
security = user
server string = 73 samba server %v
encrypt passwords = yes

interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
hosts allow = 127. 192.168.12. 192.168.13.

log file = /var/log/samba/%m.log
max log size = 50

unix password sync = Yes

time server = Yes

domain logons = Yes

logon script = logon.bat
logon path = \\%L\profiles\%u\%m
logon drive = Z:
logon home = \\%L\%u\.win_profile\%m

os level = 65

preferred master = Yes
domain master = Yes

wins server = 192.168.0.7

[netlogon]
path = /usr/local/samba/lib/netlogon
write list = ntadmin
browseable = No

[profiles]
path = /home/sampa-ntprof
read only = No
create mask = 0600
directory mask = 0700
browseable = No

[homes]
comment = Home Directories
read only = No
```

```

browseable = No

[printers]
comment = All Printers
path = /var/spool/samba
browseable = No
guest ok = No
writable = No
printable = Yes

[myshare]
comment = annet MyShare
path = /
valid users = mary fred
public = yes
writable = no
browsable = yes

```

Описание параметров:

```
[global]
```

Секция `global` содержит конфигурационные параметры сервера. Остальные секции, обозначенные именами в прямых скобках – это выделяемые ресурсы.

```
workgroup = ics-73
```

Опция ***workgroup*** определяет рабочую группу, в которую входит ваш сервер. Как сказано выше, рабочая группа и домен в данном контексте - одно и то же. Важно, чтобы клиенты и сервер входили в одну и ту же группу.

```
server string = 73 samba server %v
```

Опция ***server string*** определяет строку, которую получают пользователи в блоке комментария к принтеру в менеджере принтеров, или при IPC соединении по команде ***net view*** на Windows машинах.

```
wins server = 192.168.0.7
```

Опция ***wins server*** определяет сервер имён сети Windows, данный сервер обеспечивает просмотр ресурсов сети за пределами одного сегмента. Если не обходимо включить на данном сервере WINS, то необходима следующая строка:

```
wins support = yes
```

```
time server = yes
```

Опция *time server* определяет данный сервер как сервер времени для сети Windows, данный сервер сам должен быть синхронизирован со службами точного времени по протоколу NNTP.

```
encrypt passwords = yes
```

Опция *encrypt passwords* если установлена в *yes*, инструктирует Samba использовать зашифрованные пароли вместо паролей с открытым текстом, включена по умолчанию. Устанавливается в *no*, если в сети есть клиенты старых версий Windows (95, 98, ME).

```
interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
hosts allow = 127. 192.168.12. 192.168.13.
```

Эти опции задают ограничения на доступность сервисов только на определённых интерфейсах или в определённых сетях.

```
security = user
```

Опция *security* задает режим аутентификации и авторизации. Если она установлена в режим *user* – проверка доступа пользователя к ресурсу производится на этом сервере, если в режим *domain* – то на текущем контролере домена, если *ads* – то на сервере Active Directory. Samba использует собственную базу пользователей, а не системную, потому для доступа к ресурсам необходимо отдельно добавлять пользователей, например, командой *smbpasswd*, которая будет добавлять пользователей в ту систему авторизации, которая является текущей, естественно, при наличии прав у пользователя на такие действия.

```
log file = /var/log/samba/%m.log
```

Опция *log file* определяет месторасположение и имена файлов регистрации Samba. С расширением "%m", будут создаваться независимые файлы регистрации для каждого пользователя или машины, соединяющихся к вашему Samba серверу (например, log.machine1).

```
domain master = yes
domain logons = yes
```

Опция *domain master* определяет, что один из демонов Samba, *nmbd*, будет установлен как домен мастер для данной рабочей группы. Эта опция обычно устанавливается в *yes* только на одном сервере Samba в некоторой сети и рабочей группе. Опция *domain logons* включает режим PDC для домена NT.

```
preferred master = yes
```

Опция ***preferred master*** определяет и контролирует, является ли ***nmbd*** привилегированным (preferred) мастер браузером рабочей группы. Эта опция обычно выставляется в ***yes*** на одном сервере на вашей сети.

```
os level = 65
```

Опция ***os level*** определяет значение - имеет ли ***nmbd*** шанс стать локальным мастер браузером для рабочей группы в локальной широковещательной области. Число 65 позволит «победить» любой NT сервер. Если в вашей сети есть NT сервер и вы хотите, чтобы Linux Samba сервер стал локальным мастер браузером, вы должны установить этот параметр равным 65. Эта опция должна быть определена только на одном Linux Samba сервере в сети, а на остальных ее надо отключить.

```
unix password sync = yes
```

Используется для синхронизации базы данных паролей ОС Windows и ОС UNIX при изменении пароля у пользователя.

Далее рассмотрим выделение сетевых ресурсов. Сетевые ресурсы выделяются при помощи создания именованных секций. Например, для выделения ресурса `myshare` описываем секцию:

```
[myshare]
comment = мой дисковый ресурс
```

Опция ***comment*** позволяет вам определить комментарии, которые появляются, когда клиент организует запросы на сервер.

```
path = /home/ftp
```

Опция ***path*** определяет каталог, в который разрешается доступ сети.

```
browsable = yes
```

Сообщает серверу о том, допустим ли просмотр монтируемого ресурса. Значение ***no*** означает, что ресурс будет доступен, но пользователь не будет видеть данный ресурс в сетевом окружении.

```
valid users = mary fred
```

Опция ***valid users*** определяет список пользователей, которые могут подключаться к этому сервису, необходима для ограничения доступа пользователей к ресурсу. Если опция не указана, к ресурсу имеют доступ все

пользователи. Символ @ перед именем задает NIS или UNIX группу, + - только UNIX группу, & - только NIS.

Специальные секции задают специфические сетевые ресурсы. Секция *[homes]* задает выделение домашних каталогов пользователей как доступных по сети. Имя ресурса – это логин пользователя. Секции *[netlogon]* и *[profiles]* задают ресурса для размещения скриптов, которые выполняются на клиентских машинах при старте и место для сохранения профилей пользователей. Секция *[printers]* содержит описание разделяемых принтеров.

2.3 Ход работы

Лабораторная работа выполняется на 2-х компьютерах, один из которых работает под управлением ОС Windows, другой – под управлением ОС Linux.

Перед выполнением работы необходимо проверить, установлены ли необходимые пакеты, для этого выполните команду `rpm -qa | grep samba`.

На первом, ознакомительном этапе, необходимо запустить утилиту управления сервисами Samba Web Administration Tool – *swat* и сконфигурировать сервер в простейшем варианте – выделить домашние директории и один общий публичный ресурс.

На втором этапе необходимо изучить возможности сервера Samba и директивы конфигурации. Построение контроллера домена Active Directory не входит в данную работу.

2.3.1 Установка простого сервиса разделения ресурсов.

Современные ОС для настольных ПК имеют в составе графических утилит управления системой имеют утилиту управления сервером Samba, однако наиболее полный интерфейс управления поставляется разработчиками Samba. Обычно, его нужно устанавливать отдельно командой:

```
yum install samba-swat
```

Программа *swat* запускается как сервис через сетевой супер-демон *xinetd*. Конфигурация отдельных сервисов находится в директории */etc/xinetd.d*. Откройте файл */etc/xinetd.d/swat* в редакторе и измените следующие опции:

```
disable    = no
# bind     = 127.0.0.1
```

Первая опция разрешает сервис, вторая задает адреса, для которых сервис доступен. Если вы собираетесь запускать браузер на другом компьютере, то ее необходимо закоментировать. Далее необходимо перезапустить сервис `xinetd` командой `/etc/init.d/xinetd restart`.

Теперь нужно запустить браузер и указать следующий адрес: `http://localhost:901`. Браузер запросит параметры авторизации, необходимо ввести логин `root` и соответствующий пароль. Ниже на рисунке приведен интерфейс программы.

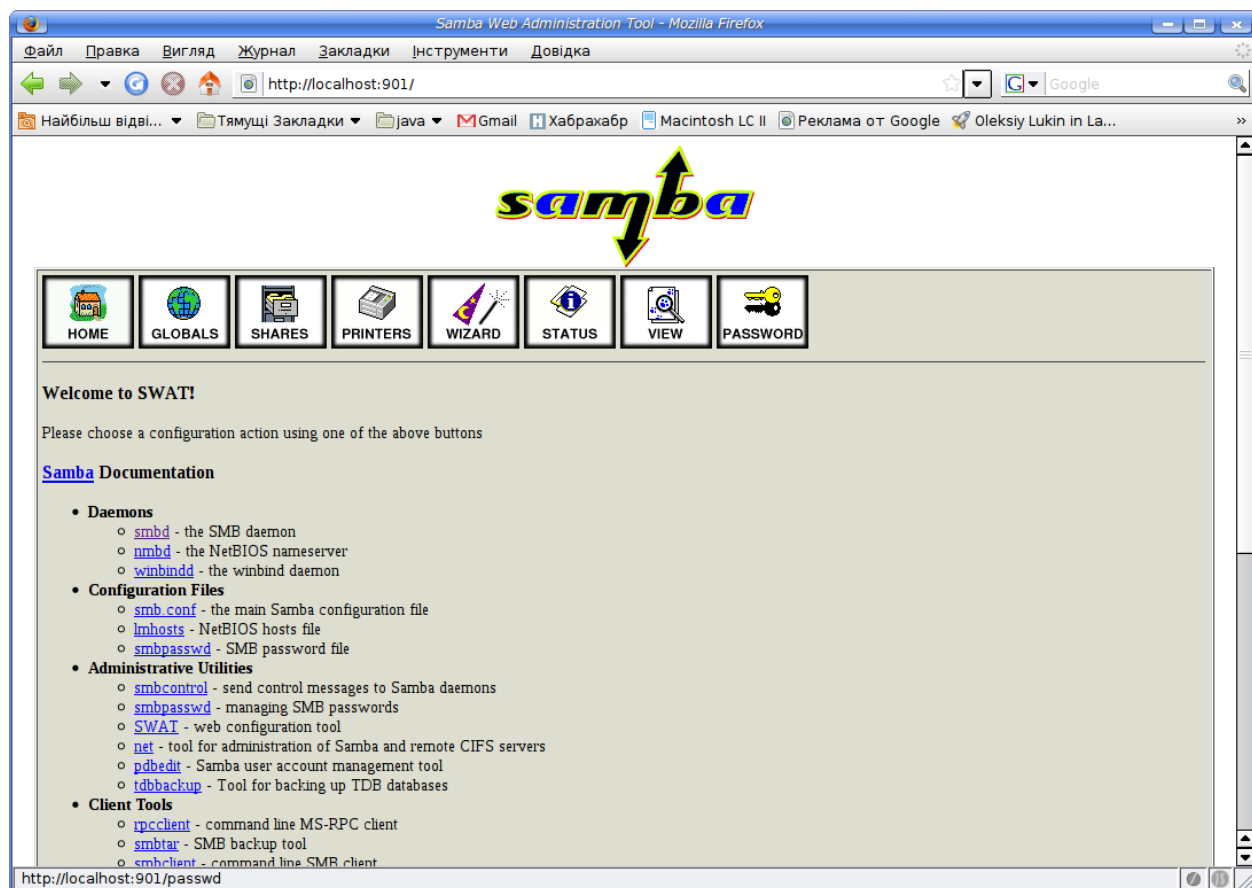


Рисунок 4. Интерфейс правления программы swat.

При удачном входе программа `swat` загружает страничку с документацией. Если установлен пакет с документацией `samba`, то все ссылки будут указывать на файлы на локальной файловой системе. Рекомендуется перед выполнением работы прочесть первую часть «Samba by Example».

Верхний набор пиктограмм используется для выбора режимов управления. Пиктограмма «Globals» выбирает страницу управления секцией `[globals]` файла конфигурации `smb.conf`, т.е. настройками сервера. Пиктограмма «Shares» выбирает страницу выделения ресурсов, «Printers» - деления принтеров. Пиктограмма «Wizard» указывает на страницу быстрой настройки сервера, пиктограмма «Status» выбирает страницу просмотра и перезапуска сервисов, пиктограмма «View» позволяет просмотреть файл

конфигурации smb.conf, и, наконец, пиктограмма «Password» уазывает на страницу управления пользователями.

Для начала выберите пиктограмму «Wizard» и настройте сервер в режим простой рабочей станции, не являющейся контроллером домена. Через пиктограмму «Shares» выделите домашние директории пользователей и один общий ресурс. Обратите внимание, что права UNIX являются определяющими, поэтому директория, в которую ползователи должны иметьправо на запись, должна иметь соответствующих владельца, группу и права. Рекомендуются завести группу smbusers и включить в нее ваших пользователей средствами ОС.

Выделите в качестве доступных ресурсов домашние каталоги пользователей и какой-нибудь общий для всех катфлог. Перезапустите сервер и проверьте выделенные ресурсы, как описано в разделе «Тестирование сервера».

Просмотрите файл конфигурации smb.conf и прочтите помощь по всем опциям, включенным в данной режиме.

2.3.2 Тестирование сервера

Для тестирования samba-сервера необходимо создать пользователя. Сначала создаем группу:

```
/usr/sbin/groupadd smbusers
```

Затем создаем пользователя:

```
useradd -c Test Samba -g smbusers -s /bin/false  
smbtester
```

Добавляем нашего тестового пользователя в список пользователей samba:

```
smbpasswd -a smbtester
```

Запуск сервера осуществляется/ командой **/sbin/service smb start**

Проверим корректность запуска сервера в системной журнале:

```
tail /var/log/messages
```

Проверяем доступность ресурсов samba-сервера сначала с локальной машины:

```
smbclient -L localhost smb_user
```

После проверки работоспособности сервера необходимо выполнить настройку рабочих станций Windows. Для этого необходимо на соседней машине загрузит операционную систему Windows, в режиме пользователя – администратора. Проверка сервера будет заключаться в подключении сетевого диска и установки системного времени, синхронизированного с samba-сервером. Желательно использовать не графическую оболочку, а оболочку cmd. Для работы с сетевыми ресурсами используется команда **net**.

Присоединение сетевых дисков осуществляется командой **net use**, например:

```
net use h: \\ics-73-10\student
```

Команда **net use** без параметров показывает текущее использование сетевых ресурсов.

```
Новые подключения будут запомнены.
Состояние   Локальный   Удаленный           Сеть
-----
OK   Y:        \\ics-73-10\rt      Microsoft Windows Network
OK   Z:        \\ics-73-10\rt      Microsoft Windows Network
Отсоединен  \\192.168.7.48\IPC$ Microsoft Windows Network
Команда выполнена успешно.
```

Команда **net view** показывает видные в сети ресурсы:

```
Общие ресурсы на \\ics-73-10
73 samba server
Имя общего ресурса  Тип      Используется как  Комментарий
-----
phenom              Диск      Home directory of phenom
rt                  Диск Z:      Mary's and Fred's stuff
Команда выполнена успешно.
```

Команда **net time** показывает, а с параметром **/set** устанавливает локальное время по серверу:

```
Текущее время на \\ics-73-10 равно 12/7/2005 12:44 PM
Команда выполнена успешно.
```

2.4 Содержание отчета

Отчет должен содержать файл конфигурации сервиса samba, а также отображать результаты всех выше указанных настроек сервиса. Наличие соответствующих комментариев и выводов необходимо.

2.5 Контрольные вопросы

Что такое сетевой сервис Samba?

Как установить пакет Samba? Какие демоны необходимы для работы Samba?

Какие настройки необходимо выполнить минимально для пакета Samba?

Как производится выделение домашних каталогов пользователей?

Как установить время на рабочей станции по серверу Samba?

Как настроить кириллицу в именах файлов и каталогов?

3 Лабораторная работа № 3. Конфигурирование службы DHCP в корпоративной сети

3.1 Цель работы

Цель данной работы – ознакомление с сервисом DHCP, а также приобретение навыков в его настройке.

3.2 Краткие теоретические сведения

3.2.1 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.

3.2.2 DHCP сервер под Unix

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).

3.2.3 DHCP сервер под Window

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).

3.2.4 DHCP-клиент под Unix

DHCP клиент 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 секунд;

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.

3.2.5 DHCP-клиент под 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.

3.3 Ход работы

Выполнение лабораторной работы состоит из следующих шагов:

1. Проверка наличия установленных пакетов `dhcp` (команда `rpm -qa | grep dhcp`)

Если пакеты установлены – приступаем к работе (шаг два), если нет – устанавливаем необходимые пакеты (`dhcpd-XXX.rpm`).

2. Создание конфигурационного файла сервера `/etc/dhcpd.conf` по приведенному выше примеру (подробности `man dhcpd.conf`)

3. Настройка одной из машин в аудитории на получение IP адреса автоматически.

Зафиксируем результаты в отчете (ifconfig /all на клиенте и /var/lib/dhcp/dhcpd.leases на сервере)

4. Создание статической записи для каждого клиента сети на основе файла /var/lib/dhcp/dhcpd.leases. Зафиксируем результат в отчет.

3.4 Содержание отчета

Отчет должен содержать конфигурационный файл сервера /etc/dhcpd.conf, а также отображать результаты всех выше указанных действий. Наличие соответствующих комментариев и выводов необходимо.

3.5 Контрольные вопросы

Что представляет собой DHCP?

Как сконфигурировать DHCP-сервер под Unix?

Как сконфигурировать DHCP-клиент под Unix?

Какие основные параметры указываются в файле конфигурации dhcpd.conf?

Опишите механизм выделения IP-адресов с помощью сетевого сервиса DHCP.

В каком случае рекомендуется выделять фиксированные адреса хостов?

Какие параметры получает рабочая станция от сервера DHCP?

ЛИТЕРАТУРА

<http://www.kernel.org/LDP> – Проект документирования Linux

<http://www.freebsd.org/handbook> – Проект документирования FreeBSD

UNIX. Пособие системного администратора./Пер. с англ. Под ред. д – К.: BHV, 2002 г.

Э. Таненбаум. Компьютерные сети./Пер. с англ. Под ред. д – К.: BHV, 2002 г.

В.Г. Олифер, Н.А. Олифер. Компьютерные сети. Принципы, технологии, протоколы. СПб., Питер, 2001-672с.:ил, ШЫИТ 5-8046-0133-4

Craig Hunt. TCP/IP network administration. O'Reilly & Associates, Inc, 1994-1998. 472 pages.

<http://www.rfc-editor.org> RFC center

<http://www.isc.org> Сайт проектов bind, dhcpd

<http://www.samba.org> Сайт проекта Samba