Система DNS.

DNS была разработана Полом Мокапетрисом в 1983 году.

В сети интернет DNS выполняет важную задачу, для доступа к веб-серверу необходимо знать его IP-адрес. Необходимость использования DNS обусловлена тем, что **людям легче запоминать буквенные (обычно осмысленные) адреса**, чем последовательность четырех цифр IP – адреса, компьютерам, в свою очередь, удобнее обрабатывать численное представление адреса (IP-адрес).

Также наличие символьного имени сервера позволяет использовать так называемые **виртуальные серверы** , например, HTTP-серверы, отличающиеся друг от друга именем запроса (доменным именем), но использующие один и тот же IP-адрес.

1. Плоские символьные имена.

В операционных системах, которые первоначально разрабатывались для локальных сетей, таких как Novell NetWare, Microsoft Windows или IBM OS/2, пользователи всегда работали с символьными именами компьютеров. Так как локальные сети состояли из небольшого числа компьютеров, применялись так называемые плоские имена, состоящие из последовательности символов, не разделенных на части.

Примерами таких имен являются: NW1_1, mail2, MOSCOW_SALES_2. Для установления соответствия между символьными именами и MAC-адресами в этих операционных системах применялся механизм широковещательных запросов, подобный механизму запросов протокола ARP. Так, широковещательный способ разрешения имен реализован в протоколе NetBIOS, на котором были построены многие локальные ОС. Так называемые NetBIOS-имена стали на долгие годы одним из основных типов плоских имен в локальных сетях.

Для стека TCP/IP, рассчитанного в общем случае на работу в больших территориально распределенных сетях, подобный подход оказывается неэффективным.

2. Иерархические символьные имена.

В стеке TCP/IP применяется доменная система имен, которая имеет иерархическую древовидную структуру, допускающую наличие в имени произвольного количества составных частей (рис. 1).

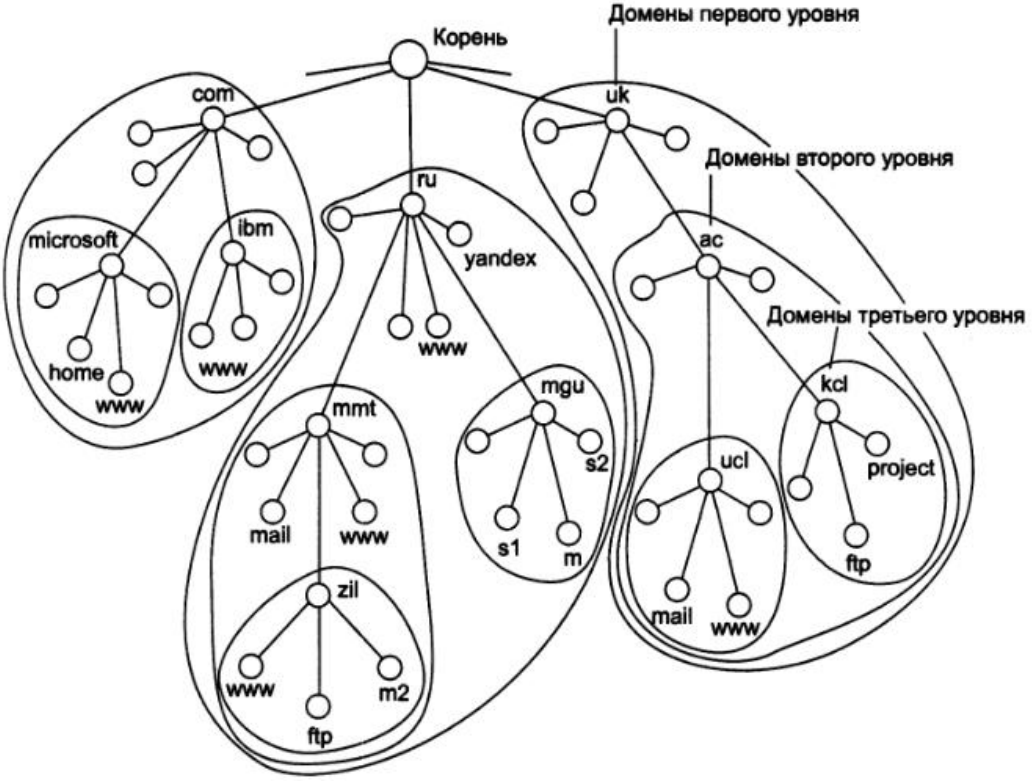


Рис. 1. Пространство доменных имен

Иерархия доменных имен аналогична иерархии имен файлов, принятой во многих популярных файловых системах. Дерево имен начинается с корня, обозначаемого здесь

точкой (.). Затем следует старшая символьная часть имени, вторая по старшинству символьная часть имени и т. д. Младшая часть имени соответствует конечному узлу сети. В отличие от имен файлов, при записи которых сначала указывается самая старшая составляющая, затем составляющая более низкого уровня и т. д., запись доменного имени начинается с самой младшей составляющей, а заканчивается самой старшей. Составные части доменного имени отделяются друг от друга точкой. Например, в имени home.microsoft.com составляющая home является именем одного из компьютеров в домене mlcrosoft.com.

Разделение имени на части позволяет разделить административную ответственность за назначение уникальных имен между различными людьми или организациями в пределах своего уровня иерархии. Так, для примера, приведенного на рис. 1, один человек может нести ответственность за то, чтобы все имена с окончанием «ru» имели уникальную следующую вниз по иерархии часть. То есть все имена типа **www.ru**, mall.mmt.ru или m2.zil.mmt. ru отличаются второй по старшинству частью.

Разделение административной ответственности позволяет решить проблему образования уникальных имен без взаимных консультаций между организациями, отвечающими за имена одного уровня иерархии. Очевидно, что должна существовать одна организация, отвечающая за назначение имен верхнего уровня иерархии.

Совокупность имен, у которых несколько старших составных частей совпадают, образуют домен имен (domain). Например, имена **www.zil.mmt.ru**, ftp.zil.mmt.ru, yandex.ru и sl.mgu. ru входят в домен ru, так как все они имеют одну общую старшую часть — имя ru.

Другим примером является домен mgu.ru. Из представленных на рис. 1 имен в него входят имена s1.mgu.ru, s2.mgu.ru и rn.mgu.ru. Этот домен образуют имена, у которых две старшие части равны mgu.ru. Администратор домена mgu.ru несет ответственность за уникальность имен следующего уровня, входящих в домен, то есть имен s1, s2 и m. Образованные домены s1.mgu.ru, s2.mgu.ru и rn.mgu.ru являются поддоменами домена mgu.ru, так как имеют общую старшую часть имени. Часто проддомены для краткости называют только младшей частью имени, то есть в нашем случае поддоменами являются s1, s2 и m.

Если в каждом домене и поддомене обеспечивается уникальность имен следующего уровня иерархии, то и вся система имен будет состоять из уникальных имен.

По аналогии с файловой системой в доменной системе имен различают краткие, относительные и полные доменные имена.

Краткое доменное имя — это имя конечного узла сети: хоста или порта маршрутизатора. Краткое имя — это лист дерева имен.

Относительное доменное имя — это составное имя, начинающееся с некоторого уровня иерархии, но не с самого верхнего. Например, **www.zil** — это относительное имя.

Полное доменное имя (Fully Qualified Domain Name, FQDN) включает составляющие всех уровней иерархии, начиная от краткого имени и кончая корневой точкой: **www.zil.mmt.ru**.

2.1. Корневой домен.

Корневой домен управляется центральными органами Интернета, в частности уже упоминавшейся нами организацией ICANN. Домены верхнего уровня назначаются для каждой страны, а также для различных типов организаций. Имена этих доменов должны следовать международному стандарту ISO 3166.

Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, например ru (Россия), uk (Великобритания), fi (Финляндия), us (Соединенные Штаты), а для различных типов организаций, например, следующие обозначения:

- com — коммерческие организации (например, microsoft.com);
- edu — образовательные организации (например, mit.edu);
- gov — правительственные организации (например, nsf.gov);
- org — некоммерческие организации (например, fidonet.org);
- net — сетевые организации (например, nsf.net).

Каждый домен администрирует отдельная организация, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Чтобы получить доменное имя, необходимо зарегистрироваться в какой-либо организации, которой делегированы полномочия по распределению имен доменов.

Доменная система имен реализована в Интернете, но она может работать и как автономная система имен в любой крупной корпоративной сети, которая хотя и использует стек TCP/IP, никак не связана с Интернетом.

2.2. Типы записей DNS

Основные типы записей используемые в протоколе DNS

- **A запись** (address record IPv4) или запись адреса - основная запись, выполняет связующую роль между именем хоста (just-networks.ru) и IP адресом (5.101.153.37). Если меняется только A запись, то это значит, что наш сайт физически будет размещен на другом хостинге, а все остальные записи останутся работать на старом хостинге.

Название	Тип записи	Адрес
just-networks.ru	A	5.101.153.37

- **AAA запись** (address record IPv6) или запись адреса - аналогична записи A, только для IPv6.

Название	Тип записи	Адрес
just-networks.ru	AAA	FFEA::CA28:1210:4362

- **CNAME запись** (canonical name record) или каноническая запись имени (псевдоним) - используется для перенаправления на другое имя (по аналогии с ссылками), частным примером использования CNAME записи, является создание доменных имен для ftp, mail, ssh, например

Название	Тип записи	Адрес
ftp.just-networks.ru	CNAME	www.just-networks.ru
mail.just-networks.ru	CNAME	www.just-networks.ru
ssh.just-networks.ru	CNAME	www.just-networks.ru

- **MX запись** (mail exchange) или почтовый обменник, указывает те сервера, с которыми будет осуществлен обмен для данного домена. То есть определяет сервер, который будет обрабатывать почту для вашего домена. В случае отсутствия MX-записи, запрашивается A-запись

Название	Тип записи	Адрес
www.just-networks.ru	MX	mx1.beget.ru
www.just-networks.ru	MX	mx2.beget.ru

- **NS запись** (name server) указывает на DNS сервер текущего домена, так называемые authoritative DNS-серверы. Смена NS-записи, при переходе на другой хостинг, всечёт за собой смену всех записей, соответственно нужно или указывать новые записи или копировать со старого сайта (например, для сохранения почты, нужно скопировать MX-запись со старого хостинга). При неправильном изменении NS записи домена, может привести к остановке работы сайта.

Название	Тип записи	Адрес
www.just-networks.ru	NS	ns1.beget.ru
www.just-networks.ru	NS	ns2.beget.ru

- **TXT запись** текстовая запись содержащая 254 байта любой текостовой информации, в основном используется для подтверждения принадлежности домена для севрисов yandex, google.

Название	Значение
yandex	validate value for yandex

3. Схема работы DNS.

3.1. В локальных сетях.

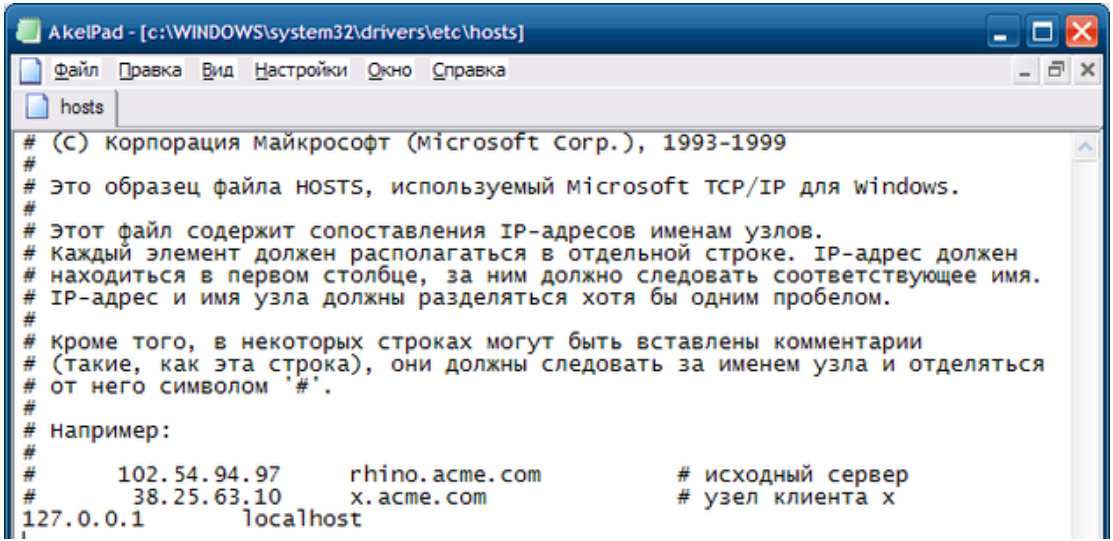
1. Широковещательный способ установления соответствия между символьными именами и локальными адресами, подобный протоколу ARP, хорошо работает только в небольшой локальной сети, не разделенной на подсети. В крупных сетях, где возможность всеобщей широковещательной рассылки не поддерживается, нужен другой способ разрешения символьных имен.

2. Хорошей альтернативой широковещательной рассылке является применение **централизованной службы**, поддерживающей соответствие между различными типами адресов всех компьютеров сети. Например, компания Microsoft для своей корпоративной операционной системы Windows NT разработала централизованную службу **WINS**, которая поддерживала базу данных NetBIOS-имен и соответствующих им IP-адресов.

3.2. В сетях TCP/IP.

Соответствие между доменными именами и IP-адресами может устанавливаться средствами как локального хоста, так и централизованной службы.

3. На раннем этапе развития Интернета на каждом хосте вручную создавался **текстовый файл** с известным именем hosts.txt. Этот файл состоял из некоторого количества строк, каждая из которых содержала одну пару «доменное имя — IP-адрес», например:



Файл hosts , расположен:

в Windows: %SystemRoot%\system32\drivers\etc\hosts;

в Unix: /etc/hosts;

По мере роста Интернета файлы hosts.txt также увеличивались в объеме, и создание масштабируемого решения для разрешения имен стало необходимостью.

4. Таким решением стала служба DNS (Domain Name System — система доменных имен), основанная **на распределенной базе** отображений «доменное имя — IP-адрес». DNS использует в своей работе серверы и клиенты. DNS-серверы поддерживают распределенную базу, а DNS-клиенты обращаются к серверам с запросами о разрешении имени на IP-адрес.

Служба DNS использует текстовые файлы почти такого же формата, как и файл hosts, и эти файлы администратор также подготавливает вручную. Однако служба DNS опирается на иерархию доменов, и каждый DNS-сервер хранит только часть имен сети, а не все имена, как это происходит при использовании файлов hosts. При росте количества узлов в сети проблема масштабирования решается созданием новых доменов и поддоменов имен и добавлением в службу DNS новых серверов.

3.3. DNS сервер.

Для каждого домена имен создается свой DNS-сервер. На серверах применяют два подхода к распределению имен.

В первом случае сервер может хранить отображения «доменное имя — IP-адрес» для всего домена, включая все его поддомены. Однако такое решение оказывается плохо масштабируемым, так как при добавлении новых поддоменов нагрузка ка этот сервер может превысить его возможности.

Чаще используется другой подход, когда сервер домена хранит только имена, которые заканчиваются на следующем ниже уровне иерархии по сравнению с именем домена. (Аналогично каталогу файловой системы, который содержит записи о файлах и подкаталогах, непосредственно в него «входящих».) Именно при такой организации службы DNS нагрузка по разрешению имен распределяется более равномерно между всеми DNS-серверами сети. Например, в первом случае DNS-сервер домена mmt.ru будет хранить отображения для всех имен, заканчивающихся на mmt.ru (www.zil.mmt.ru, ftp.zil.mmt.ru, mail.mmt.ru и т. д.). Во втором случае этот сервер хранит отображения только имен типа mail.mmt.ru, www.mmt.ru, а все остальные отображения должны храниться на DNS-сервере поддомена zil.nmt.ru.

Каждый DNS-сервер помимо таблицы отображений имен содержит IP-ссылки на DNS-серверы своих поддоменов. Эти ссылки связывают отдельные DNS-серверы в единую службу DNS.

Для обслуживания корневого домена выделено несколько дублирующих друг друга DNS-серверов, IP-адреса которых широко известны (их можно узнать, например, в InterNIC).

3.3.1. Структура файла зоны.

Файл зоны формируется из набора записей. Каждая запись имеет следующий формат:

[ИМЯ] [TTL] [CLASS] TYPE VALUE

В квадратные скобки обозначают, что в конфигурационном файле сервера BIND эти поля при некоторых обстоятельствах можно не указывать. В этом случае имя будет повторять имя предыдущей записи, TTL – сформирован по определённым (описанным далее) правилам, а класс будет иметь значение IN. Поля TYPE и VALUE являются обязательными. Поле TTL означает время в секундах, в течении которого резолвер может считать кешированное значение валидным.

3.3.2. Классы записей в зоне DNS.

Существуют классы записи. Использовались только три класса: **IN (INET)**, CH (CHAOSNET) и HS (HESIOD). Классы задумывались для того, чтобы система DNS могла работать не только в сети Интернет. В настоящее время классы CH и HS не используются.

3.3.3. Типы записей в зоне DNS.

Для каждого узла в системе DNS может быть неограниченное число записей одного типа. Существует большое множество типов записей. Наиболее известные:

- SOA – Start of Authority (см. далее) ;
- NS – nameserver – показывает, где искать имя;
- A – показывает соответствие имени IP адресу;
- AAAA – соответствие имени IPv6 адресу;
- PTR – соответствие IP адреса имени;
- MX – mail exchanger – указывает на почтовый сервер (см. SMTP);
- TXT – несёт в себе произвольную текстовую информацию;
- CNAME – canonical name – или привязка дополнительных имён к основному;
- NULL – запись позволяет передавать любые бинарные данные и используется крайне редко оторванными от реальности индивидами для построения VPN поверх системы DNS.

3.3.4. Запись SOA.

Интерес представляет запись SOA. В поле VALUE для этой записи необходимо указывать пять значений. Целиком в файле зоны это выглядит так:

```
zhmylove.ru. SOA ns.zhmylove.ru. support.zhmylove.ru. (
        2016121508 ; serial number YYYYMMDDHH
        1d         ; slave refresh
        2h         ; slave retry time in case of problems
        4w         ; slave expiration time
        1h         ; max caching time in case of failed lookups
      )
```

Первое – серийный номер зоны (2016121508). Это показатель того, нужно ли вторичному серверу запрашивать актуальное содержимое зоны. Если серийный номер записи SOA из зоны на вторичном севере меньше, чем серийный номер на первичном, вторичный запрашивает всю зону.

Второе – время запросов записи SOA вторичным сервером в обычном режиме.

Третье аналогично второму, но в режиме, когда первичный сервер не отвечает (например, неисправна сеть). Обычно, это время меньше, чтобы быстрее среагировать на восстановление первичного сервера.

Четвёртое – это время от последнего успешного трансфера зоны, через которое вторичный сервер перестаёт отвечать на запросы. Иными словами, сколько может жить зона после отключения мастера.

Пятое служит сразу для нескольких целей. Во-первых, оно означает, сколько можно для данной зоны кешировать ответ NXDOMAIN (запись не найдена). Во-вторых, это TTL по умолчанию для остальных записей, если не указана директива $TTL или оно не задаётся для записи вручную. Максимальное значение поля по стандарту – 3 часа.

3.3.5. Пример файла зоны.

Здесь мы видим обязательную запись SOA, и необходимые записи NS, имеется дополнительный сервер имен расположенный по адресу ns2.psi.net. Всегда необходимо иметь дополнительный сервер имен за пределами домена в качестве резерва. Мы видем, что этот

домен имеет основной сервер, названный land-5, который заботится о множестве разных сервисов Internet, это сделано используя записи CNAME (как альтернатива использованию записей A).

Как вы видите из записи SOA, файл зоны расположен в домене land-5.com, ответственным лицом является root@land-5.com. hostmaster -- это другой часто используемый адрес для ответственного за эту работу человека.

```
@       IN      SOA     land-5.com. root.land-5.com. (
                        199609206       ; serial, todays date + todays serial #
                        8H              ; refresh, seconds
                        2H              ; retry, seconds
                        1W              ; expire, seconds
                        1D )            ; minimum, seconds
                NS      land-5.com.
                NS      ns2.psi.net.
                MX      10 land-5.com.  ; Основной почтовый сервер

localhost       A       127.0.0.1
router          A       206.6.177.1
land-5.com.     A       206.6.177.2
ns              A       206.6.177.3
www             A       207.159.141.192
ftp             CNAME   land-5.com.
mail            CNAME   land-5.com.
```

```
news            CNAME   land-5.com.

funn            A       206.6.177.2
@               TXT     "LAND-5 Corporation"
;
;       Рабочие станции
;
ws-177200       A       206.6.177.200
ws-177201       A       206.6.177.201
ws-177202       A       206.6.177.202
ws-177203       A       206.6.177.203
```

3.4. Процедура разрешения DNS-имени.

Процедура разрешения DNS-имени во многом аналогична процедуре поиска файловой системой адреса файла по его символьному имени. Действительно, в обоих случаях составное имя отражает иерархическую структуру организации соответствующих справочников — каталогов файлов или DNS-таблиц. Здесь домен и доменный DNS-сервер являются аналогом каталога файловой системы.

Для определения IP-адреса по доменному имени необходимо просмотреть все DNS-серверы, обслуживающие цепочку поддоменов, входящих в имя хоста, начиная с корневого домена. Существенным отличием файловой системы от службы DNS является то, что первая расположена на одном компьютере, а вторая по своей природе является распределенной.

В DNS имеются следующие типы запросов: **итеративный** (он же *прямой*), **обратный** и **рекурсивный**.

Итеративный (он же *прямой*, он же *нерекурсивный*) **запрос** посылает доменное имя DNS серверу и просит вернуть либо IP адрес этого домена, либо имя DNS сервера, авторитативного для этого домена. При этом, сервер DNS не опрашивает другие серверы для получения ответа. Так работают корневые и TLD серверы.

Рекурсивный запрос посылает DNS серверу доменное имя и просит возвратить IP адрес запрошенного домена. При этом сервер может обращаться к другим DNS серверам.

Обратный запрос посылает IP и просит вернуть доменное имя.

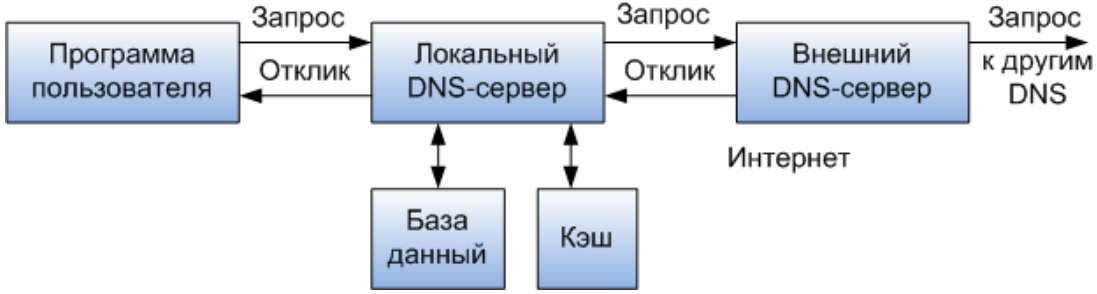
3.4.1. Итеративные запросы.

В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

1. DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени.

2. DNS-сервер отвечает клиенту, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в следующей старшей части запрошенного имени.

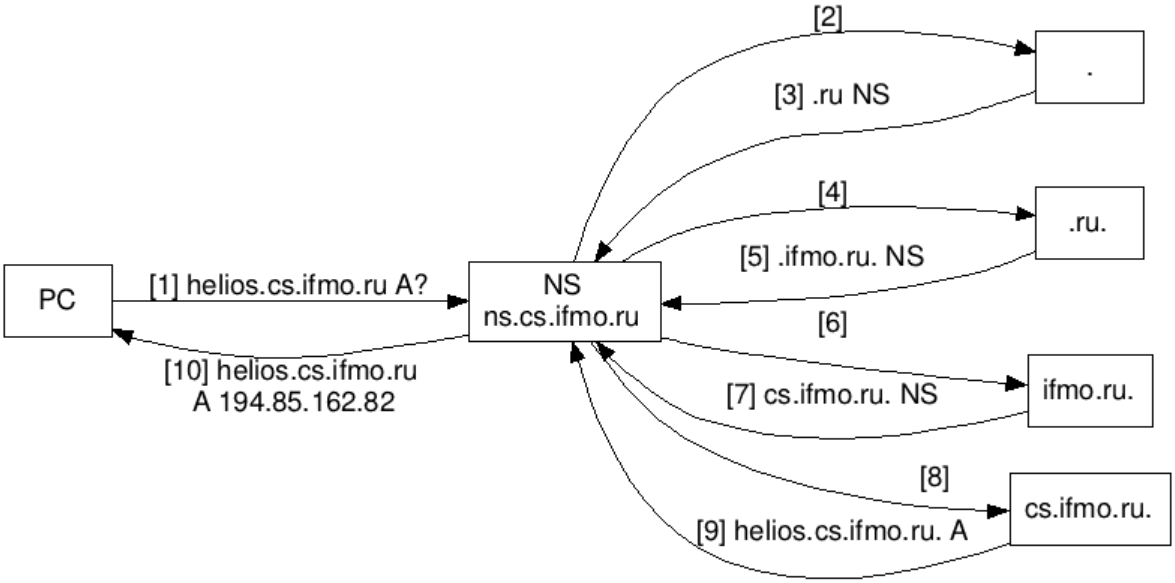
3. DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.



Такая процедура разрешения имени называется нерекурсивной, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Эта схема загружает клиента достаточно сложной работой, и она применяется редко.

3.4.2. Рекурсивные запросы.

Во втором варианте реализуется рекурсивная процедура:



1. DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, обслуживающий поддомен, которому принадлежит имя клиента.

2. Далее возможны два варианта действий:

- если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту (это может произойти, когда запрошенное имя входит в тот же поддомен, что и имя клиента, или когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше);
- если локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в предыдущем варианте, а получив ответ, передает его клиенту, который все это время просто ждет его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, именно поэтому схема называется рекурсивной, или косвенной. Практически все DNS-клиенты используют рекурсивную процедуру.

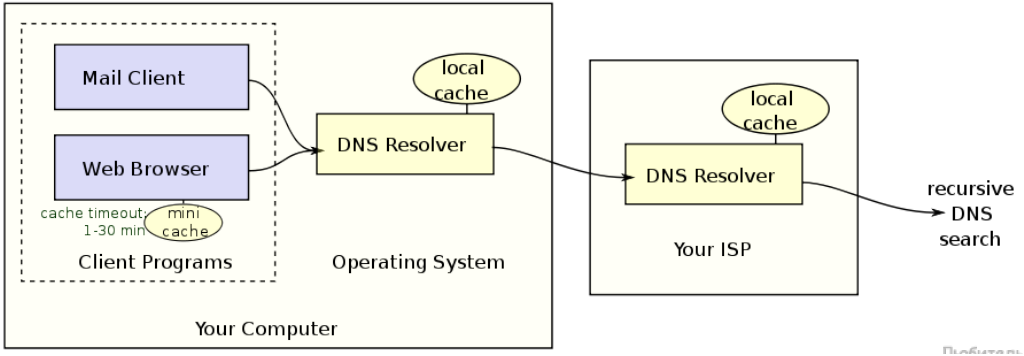
3.4.3. Кэширование.

Для ускорения поиска IP-адресов DNS-серверы широко применяют кэширование проходящих через них ответов.

Каждый, полученный системой NS, ответ кэшируется. Поэтому алгоритм обслуживания запросов не будет выполняться каждый раз, а запросы будут сразу отправляться конечному серверу или серверу доменных имён, обслуживающих известную доменную зону.

Чтобы служба DNS могла оперативно отрабатывать изменения, происходящие в сети, ответы кэшируются на относительно короткое время — обычно от нескольких часов до нескольких дней.

Кэширование проводится и на конечных хостах и даже браузерах.



4. Обратная зона.

Служба DNS предназначена не только для нахождения IP-адреса по имени хоста, но и для решения обратной задачи — нахождению DNS-имени по известному IP-адресу.

Многие программы и утилиты, пользующиеся службой DNS, пытаются найти имя узла по его адресу в том случае, когда пользователем задан только адрес (или этот адрес программа узнала из пришедшего пакета).

Обратная запись не всегда существует даже для тех адресов, для которых есть прямые записи. Ее могут просто забыть создать или же ее создание требует дополнительной оплаты. Обратная задача решается в Интернете путем организации так называемых обратных зон.

Обратная зона — это система таблиц, которая хранит соответствие между IP-адресами и DNS-имена хостов некоторой сети. Для организации распределенной службы и использования для поиска имен того же программного обеспечения, что и для поиска адресов, применяется оригинальный подход, связанный с представлением IP-адреса в виде DNS-имени.

Первый этап преобразования заключается в том, что составляющие IP-адреса интерпретируются как составляющие DNS-имени. Например, адрес 192.31.106.0 рассматривается как состоящий из старшей части, соответствующей домену 192, затем идет домен 31, в который входит домен 106.

Далее, учитывая, что при записи IP-адреса старшая часть является самой левой частью адреса, а при записи DNS-имени — самой правой, то составляющие в преобразованном адресе **указываются в обратном порядке**, то есть для данного примера — 106.31.192.

Для хранения соответствия всех адресов, начинающихся, например, с числа 192, заводится зона 192 со своими серверами имен. Для записей о серверах, поддерживающих старшие в иерархии обратные зоны, создана специальная зона in-addr.arpa, поэтому полная запись для использованного в примере адреса выглядит так:

106.31.192.in-addr.arpa.

Серверы для обратных зон используют файлы баз данных, не зависящие от файлов основных зон, в которых имеются записи о прямом соответствии тех же имен и адресов.

Такая организация данных может приводить к несогласованности, так как одно и то же соответствие вводится в файлы дважды.

В целях уменьшения объёма нежелательной корреспонденции (спама) многие почтовые серверы могут проверять наличие PTR записи для хоста, с которого происходит отправка. В этом случае PTR запись для IP адреса должна соответствовать имени отправляющего почтового сервера, которым он представляется в процессе SMTP сессии.

5. Утилиты для работы с DNS.

Для работы с системой DNS существует довольно много утилит. В Microsoft Windows – это программа nslookup, в UNIX – программы dig, nslookup, host, drill и другие.

Пример работы программы host, использованной для запроса записи типа NS для корневого домена:

```
$ host -t ns .
. name server i.root-servers.net.
. name server b.root-servers.net.
. name server m.root-servers.net.
. name server g.root-servers.net.
. name server e.root-servers.net.
```

Программа nslookup имеет интерактивный и не интерактивный варианты использования. Во втором результат показывается сразу, но возможно выполнение только одного запроса (аналогично утилите host), а в интерактивном – возможно выполнение нескольких запросов. Это можно проиллюстрировать следующим примером:

```
$ nslookup
> helios
Server:
```

```
Address:
192.168.10.1
192.168.10.1#53
Name:
Address: 192.168.10.10
> set type=soa
> gmail.com
Server:         192.168.10.1
Address:        192.168.10.1#53
Non-authoritative answer:
gmail.com
        origin = ns1.google.com
        mail addr = dns-admin.google.com
        serial = 2012061200
        refresh = 21600
        retry = 3600
        expire = 1209600
        minimum = 300
Authoritative answers can be found from:
gmail.com       nameserver = ns1.google.com.
gmail.com       nameserver = ns4.google.com.
helios.cs.ifmo.ru
gmail.com
```

gmail.com

ns1.google.com
ns2.google.com
ns3.google.com
ns4.google.com
nameserver = ns2.google.com.
nameserver = ns3.google.com.
internet address = 216.239.32.10
internet address = 216.239.34.10
internet address = 216.239.36.10
internet address = 216.239.38.10

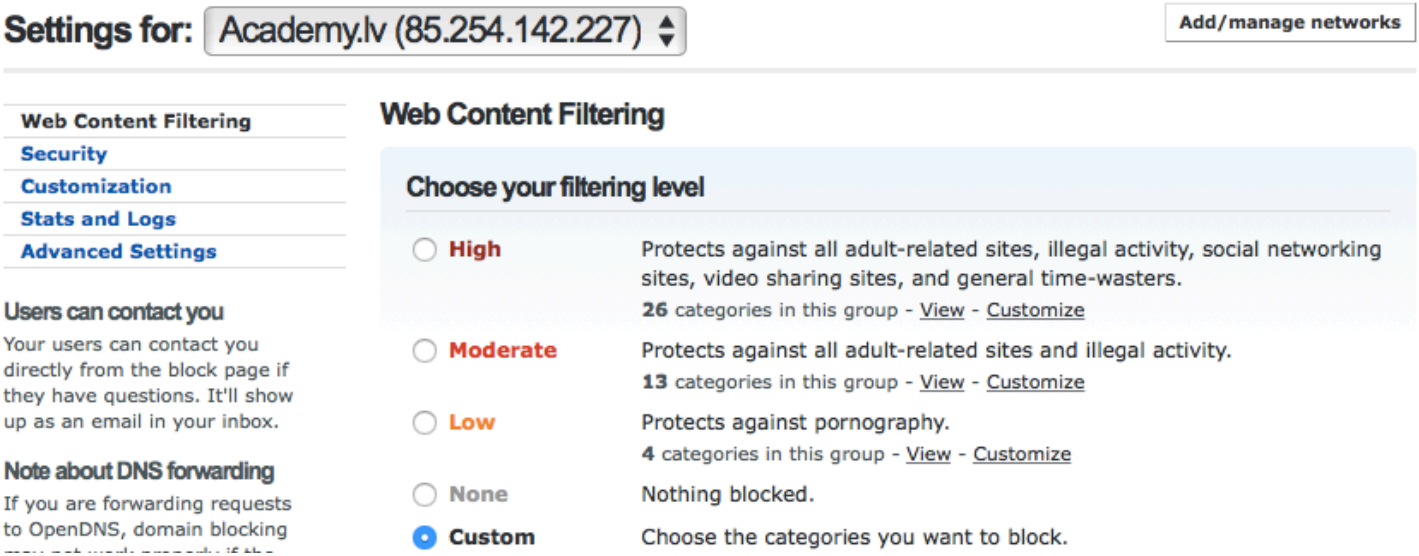
6. DNS и безопасность.

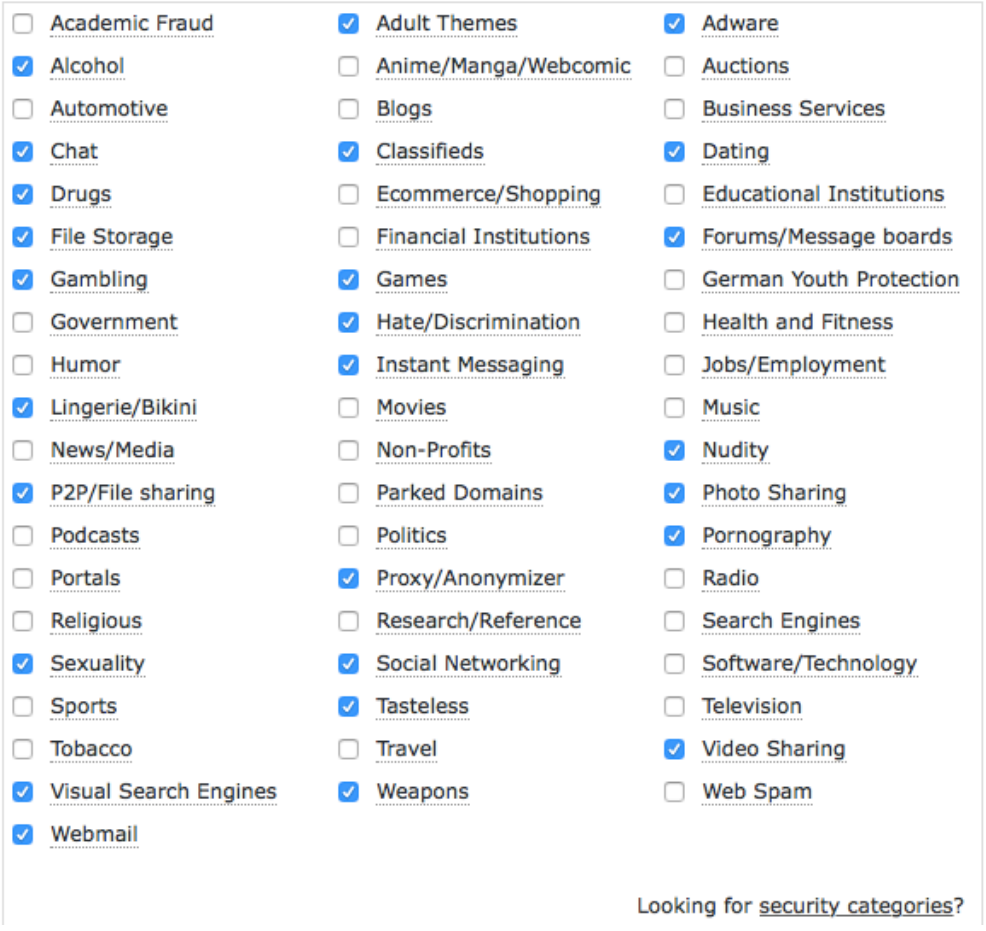
6.1. Parental Controls.

Включить "Parental Controls" можно попробовать на любом компьютере.

Для проверки задайте DNS-сервер в настройках своего компьютера или роутера 208.67.222.222 и/или 208.67.220.220. На этих DNS серверах используется фильтрация OpenDNS.

Зарегистрируйтесь на opendns.com и задайте правила фильтрации.


Settings for: Academy.lv (85.254.142.227)
Add/manage networks
Web Content Filtering
Security
Customization
Stats and Logs
Advanced Settings
Users can contact you
Your users can contact you directly from the block page if they have questions. It'll show up as an email in your inbox.
Note about DNS forwarding
If you are forwarding requests to OpenDNS, domain blocking
Web Content Filtering
Choose your filtering level
High
Protects against all adult-related sites, illegal activity, social networking sites, video sharing sites, and general time-wasters.
26 categories in this group - View - Customize
Moderate
Protects against all adult-related sites and illegal activity.
13 categories in this group - View - Customize
Low
Protects against pornography.
4 categories in this group - View - Customize
None
Nothing blocked.
Custom
Choose the categories you want to block.


Academic Fraud
Adult Themes
Adware
Alcohol
Anime/Manga/Webcomic
Auctions
Automotive
Blogs
Business Services
Chat
Classifieds
Dating
Drugs
Ecommerce/Shopping
Educational Institutions
File Storage
Financial Institutions
Forums/Message boards
Gambling
Games
German Youth Protection
Government
Hate/Discrimination
Health and Fitness
Humor
Instant Messaging
Jobs/Employment
Lingerie/Bikini
Movies
Music
News/Media
Non-Profits
Nudity
P2P/File sharing
Parked Domains
Photo Sharing
Podcasts
Politics
Pornography
Portals
Proxy/Anonymizer
Radio
Religious
Research/Reference
Search Engines
Sexuality
Social Networking
Software/Technology
Sports
Tasteless
Television
Tobacco
Travel
Video Sharing
Visual Search Engines
Weapons
Web Spam
Webmail
Looking for security categories?

6.2. DNS-spoofing.

DNS-spoofing — атака, базирующаяся на заражении кэша DNS-сервера жертвы ложной записью о соответствии DNS-имени хоста, которому жертва доверяет, и IP-адреса атакующего. Относится к числу spoofing-атак.

Может применяться как непосредственно против хоста-клиента, выполняющего DNS-запрос к кэширующему серверу, так и по отношению к серверу, путём заражения его кэша. Во втором случае обманутыми получаются все клиенты DNS-сервера, которым он отвечает данными из своего кэша. Атака основывается на двух основных методах.

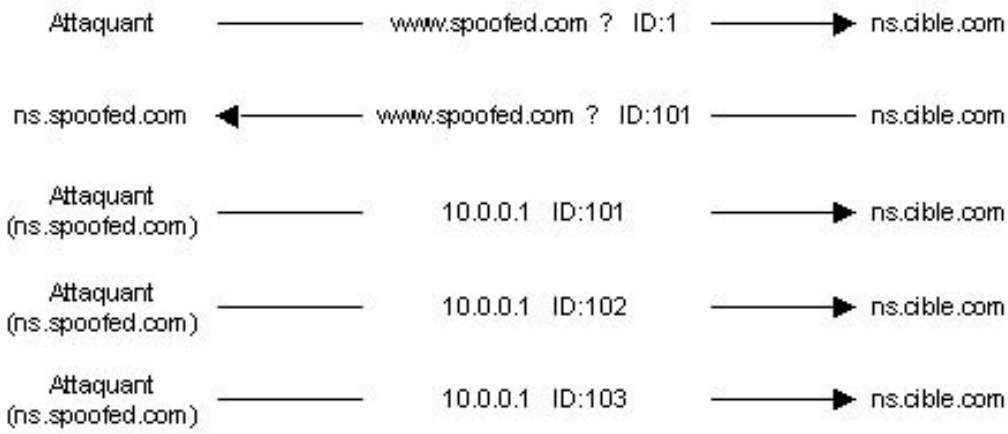
6.2.1. Подмена DNS ID (DNS ID Spoofing).

Заголовок пакета DNS-протокола содержит идентификационное поле для соответствия запросов и ответов. Целью подмены DNS ID является посылка своего ответа на DNS-запрос до того, как ответит настоящий DNS-сервер. Для выполнения этого, нужно спрогнозировать идентификатор запроса. Локально это реализуется простым прослушиванием сетевого трафика. Однако, удаленно выполнить эту задачу гораздо сложнее. Существуют различные методы:

- проверка всех доступных значений идентификационного поля. Не очень практично, поскольку общее количество возможных значений составляет 65535 (размер поля 16 бит);
- посылка нескольких сотен DNS-запросов в правильном порядке. Очевидно, что этот метод не очень надежен;

- нахождение сервера, генерирующего прогнозируемые идентификаторы (например, увеличивающиеся на 1). Этот тип уязвимости присущ некоторым версиям Bind и системам Windows 9x.

В любом случае, необходимо ответить до настоящего DNS-сервера. Этого можно достичь, например, выполнив против сервера атаку типа "отказ в обслуживании".



В результате, кэш целевого DNS-сервера будет содержать соответствие, необходимое злоумышленнику и следующим клиентам запрашивающим адрес www.spoofed.com будет сообщен адрес машины злоумышленника. На ней может быть размещена копия настоящего сайта, с помощью которого злоумышленник может красть конфиденциальную информацию.

6.2.2. DNS Cache Poisoning

DNS-сервера используют кеш для хранения результатов предыдущих запросов в течении некоторого времени. Это делается чтобы избежать постоянных повторов запросов к авторизированным серверам соответствующих доменов. Второй вариант атаки, направленной на подмену DNS, заключается в изменении кэша DNS сервера. Вот пример:

Используем те же данные, что и в предыдущем примере. Вот ключевые моменты этого варианта атаки:

- послать DNS-запрос на разрешение имени www.attaquant.com DNS-серверу домена cible.com;
- целевой DNS-сервер шлет запрос на разрешение имени www.attaquant.com DNS-серверу злоумышленника;
- DNS-сервер злоумышленника шлет ответ, содержащий несколько записей, в том числе и сфальсифицированные записи, что позволяет задавать имени соответствие с IP-адресом злоумышленника. Например, сайт www.cible.com может иметь фальсифицированную DNS-запись, соответствующую IP-адресу сайта www.attaquant.com.